

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 780 686**

51 Int. Cl.:

H04N 19/70 (2014.01)

H04N 19/30 (2014.01)

H04N 19/463 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **26.09.2014 PCT/US2014/057719**

87 Fecha y número de publicación internacional: **02.04.2015 WO15048443**

96 Fecha de presentación y número de la solicitud europea: **26.09.2014 E 14782051 (8)**

97 Fecha y número de publicación de la concesión europea: **25.12.2019 EP 3050299**

54 Título: **Tipo de dependencia entre vistas en MV-HEVC**

30 Prioridad:

27.09.2013 US 201361883931 P
25.09.2014 US 201414496807

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
26.08.2020

73 Titular/es:

QUALCOMM INCORPORATED (100.0%)
5775 Morehouse Drive
San Diego, CA 92121-1714, US

72 Inventor/es:

CHEN, YING y
WANG, YE-KUI

74 Agente/Representante:

FORTEA LAGUNA, Juan José

ES 2 780 686 T3

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín Europeo de Patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre Concesión de Patentes Europeas).

DESCRIPCIÓN

Tipo de dependencia entre vistas en MV-HEVC

- 5 **[0001]** Esta solicitud reivindica el beneficio de la solicitud provisional estadounidense número 61/883.931 presentada el 27 de septiembre de 2013.

CAMPO TÉCNICO

- 10 **[0002]** Esta divulgación se refiere a la codificación de vídeo y, más en particular, a la señalización del tipo de dependencia entre vistas en la codificación multicapa.

ANTECEDENTES

- 15 **[0003]** Las capacidades de vídeo digital se pueden incorporar a una amplia gama de dispositivos, que incluye televisores digitales, sistemas de radiodifusión digital directa, sistemas de radiodifusión inalámbrica, asistentes digitales personales (PDA), ordenadores portátiles o de escritorio, tabletas electrónicas, lectores de libros electrónicos, cámaras digitales, dispositivos de grabación digital, reproductores de medios digitales, dispositivos de videojuegos, consolas de videojuegos, teléfonos celulares o de radio por satélite, los denominados "teléfonos inteligentes", dispositivos de videoconferencia, dispositivos de transmisión continua de vídeo y similares. Los dispositivos de vídeo digital implementan técnicas de compresión de vídeo, tales como las descritas en las normas definidas por MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Parte 10, Codificación de vídeo avanzada (AVC), la norma de codificación de vídeo de alta eficacia (HEVC) actualmente en desarrollo y las ampliaciones de dichas normas. Los dispositivos de vídeo pueden transmitir, recibir, codificar, descodificar y/o almacenar información de vídeo digital con más eficacia implementando dichas técnicas de compresión de vídeo.

- 20 **[0004]** Las técnicas de compresión de vídeo realizan predicción espacial (intraimagen) y/o predicción temporal (interimagen) para reducir o eliminar la redundancia intrínseca a las secuencias de vídeo. Para la codificación de vídeo basada en bloques, un fragmento de vídeo (es decir, una trama de vídeo o una parte de una trama de vídeo) se puede dividir en bloques de vídeo, que también se pueden denominar bloques arbolados, unidades de codificación (CU) y/o nodos de codificación. Los bloques de vídeo en un fragmento intracodificado (I) de una imagen se codifican usando predicción espacial con respecto a muestras de referencia en bloques contiguos en la misma imagen. Los bloques de vídeo en un fragmento intercodificado (P o B) de una imagen pueden usar predicción espacial con respecto a muestras de referencia de bloques contiguos en la misma imagen o predicción temporal con respecto a muestras de referencia en otras imágenes de referencia. Las imágenes pueden denominarse tramas y las imágenes de referencia pueden denominarse tramas de referencia.

- 30 **[0005]** La predicción espacial o temporal da como resultado un bloque predictivo para un bloque a codificar. Los datos residuales representan diferencias de píxeles entre el bloque original a codificar y el bloque predictivo. Un bloque intercodificado se codifica de acuerdo con un vector de movimiento que apunta a un bloque de muestras de referencia que forman el bloque predictivo, y los datos residuales que indican la diferencia entre el bloque codificado y el bloque predictivo. Un bloque intracodificado se codifica de acuerdo con un modo de intracodificación y de acuerdo con datos residuales. Para una mayor compresión, los datos residuales pueden transformarse desde el dominio del píxel a un dominio de transformada, dando como resultado coeficientes de transformada residuales, que a continuación se pueden cuantificar. Los coeficientes de transformada cuantificados, dispuestos inicialmente en una formación bidimensional, pueden explorarse con el fin de generar un vector unidimensional de coeficientes de transformada, y puede aplicarse codificación por entropía para lograr aún más compresión.

- 35 **[0006]** Chen J *et al.*, en "Description of scalable video coding technology proposal by Qualcomm", analizan la predicción entre capas en la sección 2.3 de este documento.

- 40 **[0007]** Ikai *et al.* en "MV-HEVC/SHVC HLS: On dependency type" JCTVC-N0058 propone que `direct_dependency_type` igual a 0 debería representar tanto la representación de muestra como de movimiento, de modo que `direct_dependency_type` pueda excluirse estableciendo `direct_dep_type_len` igual a 0.

BREVE DESCRIPCIÓN

- 45 **[0008]** De acuerdo con las técnicas de esta divulgación, la sobrecarga de señalización asociada con la señalización de los tipos de dependencia entre capas para combinaciones de capas de datos de vídeo multicapa puede reducirse incluyendo, en un flujo de bits de datos de vídeo, un elemento sintáctico que represente si se señala un elemento sintáctico de tipo de dependencia para cada combinación de capa. En respuesta al valor para el primer elemento sintáctico que indica que el elemento sintáctico del tipo de dependencia no está señalado, un descodificador de vídeo determina que el tipo de dependencia para todas las combinaciones de capas es un tipo predeterminado y descodifica un bloque de la capa actual usando predicción entre capas conforme al tipo predeterminado.

[0009] En un ejemplo, un procedimiento para descodificar datos de vídeo multicapa incluye recibir un valor para un primer elemento sintáctico que representa si se señala un elemento sintáctico de tipo de dependencia para una capa actual, en el que el elemento sintáctico de tipo de dependencia identifica un tipo de dependencia de una capa actual en relación con una capa de referencia; y en respuesta al valor para el primer elemento sintáctico que indica que el elemento sintáctico de tipo de dependencia no está señalado, determinando que el tipo de dependencia de la capa actual en relación con la capa de referencia es un tipo predeterminado y descodificando un bloque de la capa actual usando predicción entre capas conforme al tipo predeterminado.

[0010] En otro ejemplo, un procedimiento de codificación de datos de vídeo multicapa incluye, para datos de vídeo codificados que comprenden múltiples capas, determinar si se codifican las múltiples capas usando un mismo tipo de dependencia para todas las capas de las múltiples capas o múltiples tipos de dependencia, en el que un tipo de dependencia identifica un tipo de dependencia de una capa actual en relación con una capa de referencia; y en base a determinar si se codifican las múltiples capas usando el mismo tipo de dependencia o múltiples tipos de dependencia, determinando un valor para un primer elemento sintáctico que represente si se señala un elemento sintáctico de tipo de dependencia para una capa actual.

[0011] En otro ejemplo, un aparato que realiza codificación de vídeo incluye una memoria que almacena datos de vídeo y un codificador de vídeo que comprende uno o más procesadores configurados para: recibir un valor para un primer elemento sintáctico que representa si se señala un elemento sintáctico de tipo de dependencia para una capa actual, en el que el elemento sintáctico de tipo de dependencia identifica un tipo de dependencia de una capa actual en relación con una capa de referencia; y en respuesta al valor para el primer elemento sintáctico que indica que el elemento sintáctico de tipo de dependencia no está señalado, determinando que el tipo de dependencia de la capa actual en relación con la capa de referencia es un tipo predeterminado y descodificando un bloque de la capa actual usando predicción entre capas conforme al tipo predeterminado.

[0012] En otro ejemplo, un medio de almacenamiento legible por ordenador que almacena instrucciones que cuando son ejecutadas por uno o más procesadores hacen que uno o más procesadores reciban un valor para un primer elemento sintáctico que representa si se señala un elemento sintáctico de tipo de dependencia para una capa actual, en el que el elemento sintáctico de tipo de dependencia identifica un tipo de dependencia de una capa actual en relación con una capa de referencia; en respuesta al valor para el primer elemento sintáctico que indica que el elemento sintáctico del tipo de dependencia no está señalado, determinan que el tipo de dependencia de la capa actual en relación con la capa de referencia es un tipo predeterminado y descodifican un bloque de la capa actual usando predicción entre capas conforme al tipo predeterminado.

[0013] Un aparato para descodificar datos de vídeo multicapa, que incluye medios para recibir un valor para un primer elemento sintáctico que representa si se señala un elemento sintáctico de tipo de dependencia para una capa actual, en el que el elemento sintáctico de tipo de dependencia identifica un tipo de dependencia de una capa actual en relación con una capa de referencia; medios para determinar que el tipo de dependencia de la capa actual con respecto a la capa de referencia es un tipo predeterminado en respuesta al valor para el primer elemento sintáctico que indica que el elemento sintáctico de tipo de dependencia no está señalado; y medios para descodificar un bloque de la capa actual usando predicción entre capas conforme al tipo predeterminado.

[0014] Un aparato para codificar datos de vídeo que comprende, para datos de vídeo codificados que comprenden múltiples capas, medios para determinar si se codifican las múltiples capas usando un mismo tipo de dependencia para todas las capas de las múltiples capas o múltiples tipos de dependencia, en el que un tipo de dependencia identifica un tipo de dependencia de una capa actual en relación con una capa de referencia; y basado en determinar si se codifican las múltiples capas usando el mismo tipo de dependencia o múltiples tipos de dependencia, medios para determinar un valor para un primer elemento sintáctico que representa si se señala un elemento sintáctico de tipo de dependencia para una capa actual.

[0015] Los detalles de uno o más ejemplos se exponen en los dibujos adjuntos y en la descripción siguiente. Otras características, objetos y ventajas resultarán evidentes a partir de la descripción y de los dibujos, y a partir de las reivindicaciones. La invención se expone en las reivindicaciones adjuntas.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

[0016]

La FIG. 1 es un diagrama de bloques que ilustra un sistema de codificación y descodificación de vídeo ejemplar que puede usar las técnicas descritas en esta divulgación.

La FIG. 2 es un diagrama gráfico que ilustra un orden de codificación o descodificación de múltiples vistas ejemplar, de acuerdo con uno o más ejemplos descritos en esta divulgación.

La FIG. 3 muestra una estructura de predicción entre vistas y temporal de MVC ejemplar.

La FIG. 4 es un diagrama de bloques que ilustra un codificador de vídeo ejemplar que puede implementar las técnicas descritas en esta divulgación.

5 La FIG. 5 es un diagrama de bloques que ilustra un descodificador de vídeo ejemplar que puede implementar las técnicas descritas en esta divulgación.

La FIG. 6 es un diagrama de bloques que ilustra un conjunto ejemplar de dispositivos que forman parte de una red.

10 La FIG. 7 es un diagrama de flujo que ilustra un procedimiento ejemplar de codificación de datos de vídeo de acuerdo con las técnicas de esta divulgación.

La FIG. 8 es un diagrama de flujo que ilustra un procedimiento ejemplar de descodificación de datos de vídeo de acuerdo con las técnicas de esta divulgación.

15 DESCRIPCIÓN DETALLADA

[0017] Esta divulgación introduce técnicas relacionadas con la señalización del tipo de dependencia entre capas en la codificación de vídeo multicapa, tal como en el estándar emergente de codificación de vídeo de alta eficiencia de múltiples vistas (MV-HEVC). En el estándar de codificación de vídeo de alta eficiencia (HEVC), una capa en general se refiere al conjunto de capa de codificación de vídeo (VCL), unidades de capa de abstracción de red (NAL) que todas tienen (es decir, cada una comparten) un valor de identificación de capa particular (por ejemplo, el valor "nuh_layer_id" en HEVC, donde "nuh" se refiere a la "cabecera de unidad de NAL") y las unidades de NAL no VCL asociadas con la capa. Una capa puede, por ejemplo, corresponder a una vista al codificar datos de vídeo de múltiples vistas o diversas dimensiones escalables (por ejemplo, resolución espacial, PSNR, profundidad de bits de color u otras dimensiones similares) para la codificación de vídeo escalable. Cada capa de una transmisión de vídeo está asociada con un valor de identificación de capa. Por tanto, diferentes capas están asociadas con diferentes valores de identificación de capa.

[0018] La predicción entre capas en general se refiere a la predicción (por ejemplo, predecir un bloque de vídeo a partir de un bloque de vídeo ya codificado) que depende de elementos de datos, tal como valores de muestra o información de movimiento, de imágenes de referencia en una capa diferente a la de la capa (por ejemplo, una capa actual) asociada con el bloque que se predice. Una capa actual puede tener un tipo específico de dependencia entre capas en relación con una capa de referencia. El tipo de dependencia entre capas define la manera en que la capa actual puede depender de la capa de referencia. Los ejemplos de tipos de dependencia entre capas incluyen predicción de muestra entre capas, predicción de movimiento entre capas y una combinación de predicción de muestra entre capas y predicción de movimiento entre capas. Cuando la predicción de muestra entre capas está habilitada, un codificador de vídeo puede predecir un bloque actual de datos de vídeo basado en un bloque en una imagen de referencia de una capa diferente (es decir, una capa de referencia). El codificador de vídeo puede identificar el bloque en la capa diferente usando, por ejemplo, un vector de movimiento de disparidad. El vector de movimiento de disparidad puede, por ejemplo, determinarse para el bloque actual en base a la información de movimiento de bloques vecinos espaciales o temporales del bloque actual, que son bloques en la misma capa que el bloque actual.

[0019] Cuando la predicción de movimiento entre capas está habilitada, un codificador de vídeo puede predecir información de movimiento para un bloque actual basado en la información de movimiento de un bloque en una capa diferente. La información de movimiento puede, por ejemplo, incluir un vector de movimiento (por ejemplo, un vector de movimiento de disparidad o un vector de movimiento temporal), y en algunos casos, la información de movimiento también puede incluir información adicional, tal como un índice de imagen de referencia. Usando la información de movimiento del bloque en la otra capa (es decir, una capa distinta de la capa asociada con el bloque actual), el codificador de vídeo puede determinar la información de movimiento para el bloque actual y, en base a la información de movimiento determinada, puede identificar un bloque predictivo en una imagen de referencia de la misma capa que el bloque actual. Cuando se habilita una combinación de predicción de muestra entre capas y predicción de movimiento entre capas, un codificador de vídeo puede realizar tanto predicción de muestra entre capas como predicción de movimiento entre capas.

[0020] De acuerdo con algunas implementaciones de señalización de tipos de dependencia entre capas, un codificador de vídeo codifica un elemento sintáctico (por ejemplo, el elemento sintáctico "direct_dependency_type" en HEVC) para cada combinación de capas en el flujo de bits para identificar un tipo de dependencia entre capas para la combinación. Dado que cada instancia del elemento sintáctico es de dos o más bits, la señalización de los tipos de dependencia entre capas puede producir una sobrecarga de señalización indeseable, en particular para flujos de bits con un gran número de capas. De acuerdo con las técnicas de esta divulgación, una de predicción de movimiento entre capas o predicción de muestra entre capas puede deshabilitarse, reduciendo o eliminando por tanto las instancias en las que el codificador de vídeo necesita señalar un tipo de dependencia entre capas. Un codificador de vídeo puede indicarle a un descodificador de vídeo que se ha deshabilitado un tipo particular de predicción entre capas.

[0021] Deshabilitar una de predicción de muestra entre capas y predicción de movimiento entre capas puede ser funcionalmente equivalente a habilitar una de predicción de muestra entre capas y predicción de movimiento entre capas. Por ejemplo, con tres tipos de dependencia entre capas (por ejemplo, solo predicción de muestra entre capas, solo predicción de movimiento entre capas y la combinación de predicción de muestra entre capas y de movimiento entre capas), si el descodificador de vídeo deshabilita la predicción de muestra entre capas, a continuación el descodificador de vídeo deshabilita tanto la predicción de muestra entre capas solo como la combinación de predicción de muestra entre capas y de movimiento entre capas, permitiendo de este modo solo la predicción de movimiento entre capas. De forma similar, si el descodificador de vídeo deshabilita la predicción de movimiento entre capas, a continuación el codificador de vídeo habilita la predicción de muestra entre capas.

[0022] De acuerdo con un ejemplo de las técnicas de esta divulgación, un descodificador de vídeo puede recibir un valor para un primer elemento sintáctico (por ejemplo, el elemento sintáctico "direct_dep_type_len_minus2" analizado en detalle a continuación) que representa (es decir, es indicativo de) si se señala un segundo elemento sintáctico (es decir, un elemento sintáctico de tipo de dependencia) para una capa actual. El elemento sintáctico de tipo de dependencia puede identificar un tipo de dependencia de una capa actual en relación con una capa de referencia. En respuesta al valor para el primer elemento sintáctico que indica que el elemento sintáctico de tipo de dependencia no está señalado, el descodificador de vídeo puede determinar que el tipo de dependencia de la capa actual con respecto a la capa de referencia es un tipo predeterminado y descodificar un bloque de la capa actual usando predicción entre capas conforme al tipo predeterminado.

[0023] En algunas implementaciones, el tipo predeterminado puede configurarse (por ejemplo, almacenarse localmente en una memoria) en el codificador y descodificador de vídeo, por ejemplo, como parte de un códec de vídeo. En dicha configuración, el primer elemento sintáctico puede ser un solo bit, donde un primer valor (por ejemplo, un valor de "0" o un valor de "1") del primer elemento sintáctico indica que el tipo de dependencia entre capas para todas las capas es el tipo predeterminado. Por tanto, en respuesta a recibir un primer valor para el primer elemento sintáctico, el descodificador de vídeo puede determinar, sin recibir una instancia del segundo elemento sintáctico, que el tipo de dependencia entre capas para todas las combinaciones de capas de un flujo de bits de vídeo es el tipo de dependencia entre capas predeterminado definido por (o almacenado localmente en) el códec de vídeo. Un segundo valor (por ejemplo, un valor de "1" o un valor de "0") para el primer elemento sintáctico puede indicar que los tipos de dependencia entre capas se señalarán por separado para cada combinación de capas. Por tanto, en respuesta a la recepción de un segundo valor para el primer elemento sintáctico, el descodificador de vídeo puede determinar los tipos de dependencia entre capas para combinaciones de capas recibiendo (es decir, en base a la recepción) una instancia del segundo elemento sintáctico para cada combinación de capas.

[0024] En otras implementaciones, el tipo predeterminado puede señalizarse desde el codificador de vídeo al descodificador de vídeo, por ejemplo, como parte del primer elemento sintáctico. En dicha configuración, el primer elemento sintáctico puede ser un elemento sintáctico de múltiples bits, donde un primer valor para el primer elemento sintáctico indica que el tipo de dependencia entre capas para todas las combinaciones de capas es la predicción de muestra entre capas y un segundo valor para el primer elemento sintáctico indica que el tipo de dependencia entre capas para todas las combinaciones de capas es la predicción de movimiento entre capas. Por tanto, en respuesta a recibir el primer valor o el segundo valor para el primer elemento sintáctico, el descodificador de vídeo puede determinar, sin recibir una instancia del segundo elemento sintáctico, que el tipo de dependencia entre capas para todas las combinaciones de capas de un flujo de bits de vídeo es la dependencia entre capas predeterminada indicada por el primer elemento sintáctico. Un tercer valor para el primer elemento sintáctico puede indicar que los tipos de dependencia entre capas se señalarán por separado para cada combinación de capas. Por tanto, en respuesta a la recepción del tercer valor para el primer elemento sintáctico, el descodificador de vídeo puede determinar los tipos de dependencia entre capas para combinaciones de capas recibiendo (en base a la recepción) una instancia del segundo elemento sintáctico para cada combinación de capas.

[0025] De acuerdo con las técnicas de esta divulgación, cuando la predicción de movimiento entre capas o la predicción de muestra entre capas está deshabilitada, se puede configurar un descodificador de vídeo para determinar, sin ninguna señalización adicional (por ejemplo, sin recibir instancias del segundo elemento sintáctico descrito anteriormente), un tipo de dependencia entre capas para las combinaciones de capas. Por ejemplo, si la predicción de movimiento entre capas está deshabilitada, entonces el descodificador de vídeo puede configurarse para determinar, sin señalización adicional para una combinación de capas, que un tipo de dependencia entre capas para la combinación de capas es la predicción de muestra entre capas. Si bien las técnicas existentes siempre pueden codificar múltiples instancias del segundo elemento sintáctico (es decir, el elemento sintáctico de tipo de dependencia), las técnicas de esta divulgación pueden, en algunos escenarios de codificación, solo codificar una (es decir, una sola) instancia del primer elemento sintáctico y evitar (es decir, abstenerse de y/o no necesitar) codificar múltiples instancias del segundo elemento sintáctico. Al reducir o eliminar la necesidad de señalar un tipo de dependencia entre capas (por ejemplo, el segundo elemento sintáctico descrito anteriormente) para algunas combinaciones de capas, las técnicas de esta divulgación pueden, en algunos escenarios de codificación, reducir la sobrecarga de señalización asociada con la señalización de tipos de dependencia entre capas.

[0026] Como se usa en esta divulgación, el término bloque actual en general se refiere a un bloque que se está codificando actualmente, a diferencia de un bloque ya codificado o un bloque que aún no se ha codificado. Como se usa en esta divulgación, el término codificación puede usarse genéricamente para referirse a la codificación o decodificación de vídeo.

[0027] La FIG. 1 es un diagrama de bloques que ilustra un sistema ejemplar de codificación y decodificación de vídeo 10 que puede usar las técnicas de señalización de tipo de dependencia entre capas descritas en esta divulgación. Como se muestra en la FIG. 1, el sistema 10 incluye un dispositivo de origen 12 que genera datos de vídeo codificados, incluyendo datos de vídeo multicapa que se van a decodificar en un momento posterior por un dispositivo de destino 14. El dispositivo de origen 12 y el dispositivo de destino 14 pueden comprender cualquiera de una amplia gama de dispositivos, incluyendo ordenadores de escritorio, ordenadores plegables (es decir, portátiles), ordenadores de tableta, decodificadores, equipos telefónicos de mano tales como los denominados teléfonos "inteligentes", los denominados paneles "inteligentes", televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, dispositivos de transmisión continua de vídeo o similares. En algunos casos, el dispositivo de origen 12 y el dispositivo de destino 14 pueden estar equipados para la comunicación inalámbrica.

[0028] El dispositivo de destino 14 puede recibir los datos de vídeo codificados que se van a decodificar por medio de un enlace 16. El enlace 16 puede comprender cualquier tipo de medio o dispositivo que puede transferir los datos de vídeo codificados desde el dispositivo de origen 12 al dispositivo de destino 14. En un ejemplo, el enlace 16 puede comprender un medio de comunicación para posibilitar que el dispositivo de origen 12 transmita datos de vídeo codificados directamente a un dispositivo de destino 14 en tiempo real. Los datos de vídeo codificados se pueden modular de acuerdo con una norma de comunicación, tal como un protocolo de comunicación inalámbrica, y transmitir al dispositivo de destino 14. El medio de comunicación puede comprender cualquier medio de comunicación inalámbrica o por cable, tal como un espectro de radiofrecuencia (RF) o una o más líneas físicas de transmisión. El medio de comunicación puede formar parte de una red basada en paquetes, tal como una red de área local, una red de área amplia o una red global, tal como Internet. El medio de comunicación puede incluir enrutadores, conmutadores, estaciones base o cualquier otro equipo que pueda ser útil para facilitar la comunicación desde el dispositivo de origen 12 al dispositivo de destino 14.

[0029] De forma alternativa, los datos codificados se pueden emitir desde la interfaz de salida 22 a un dispositivo de almacenamiento 17. De forma similar, una interfaz de entrada puede acceder a los datos codificados del dispositivo de almacenamiento 17. El dispositivo de almacenamiento 17 puede incluir cualquiera de una variedad de medios de almacenamiento de datos de acceso distribuido o local, tales como una unidad de disco duro, discos Blu-ray, DVD, CD-ROM, una memoria flash, una memoria volátil o no volátil o cualquier otro medio de almacenamiento digital adecuado para almacenar datos de vídeo codificados. En otro ejemplo, el dispositivo de almacenamiento 17 puede corresponder a un servidor de archivos o a otro dispositivo de almacenamiento intermedio que puede contener el vídeo codificado generado por el dispositivo de origen 12. El dispositivo de destino 14 puede acceder a datos de vídeo almacenados en el dispositivo de almacenamiento 17 por medio de transmisión continua o descarga. El servidor de archivos puede ser cualquier tipo de servidor que pueda almacenar datos de vídeo codificados y transmitir esos datos de vídeo codificados al dispositivo de destino 14. Servidores de archivos de ejemplo incluyen un servidor web (por ejemplo, para un sitio web), un servidor FTP, dispositivos de almacenamiento conectados en red (NAS) o una unidad de disco local. El dispositivo de destino 14 puede acceder a los datos de vídeo codificados a través de cualquier conexión de datos estándar, incluyendo una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión inalámbrica de red de área local (WLAN)), una conexión cableada (por ejemplo, DSL, módem de cable, etc.), o una combinación de ambos que sea adecuada para acceder a los datos de vídeo codificados almacenados en un servidor de archivos. La transmisión de datos de vídeo codificados desde el dispositivo de almacenamiento 17 puede ser una transmisión continua, una transmisión de descarga o una combinación de ambas.

[0030] Las técnicas de la presente divulgación no están limitadas necesariamente a aplicaciones o configuraciones inalámbricas. Las técnicas se pueden aplicar a la codificación de vídeo como apoyo a cualquiera de una variedad de aplicaciones multimedia, tales como radiodifusiones de televisión por aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones continuas de vídeo, por ejemplo, a través de Internet, codificación de vídeo digital para su almacenamiento en un medio de almacenamiento de datos, decodificación de vídeo digital almacenado en un medio de almacenamiento de datos, u otras aplicaciones. En algunos ejemplos, el sistema 10 se puede configurar para admitir una transmisión de vídeo unidireccional o bidireccional para admitir aplicaciones tales como transmisión continua de vídeo, reproducción de vídeo, radiodifusión de vídeo y/o videotelefonía.

[0031] En el ejemplo de la FIG. 1, el dispositivo de origen 12 incluye una fuente de vídeo 18, un codificador de vídeo 20 y una interfaz de salida 22. En algunos casos, la interfaz de salida 22 puede incluir un modulador/desmodulador (módem) y/o un transmisor. En el dispositivo de origen 12, la fuente de vídeo 18 puede incluir una fuente tal como un dispositivo de captación de vídeo, por ejemplo, una videocámara, un archivo de vídeo que contiene vídeo captado previamente, una interfaz de alimentación de vídeo para recibir vídeo desde un proveedor de contenido de vídeo y/o un sistema de gráficos por ordenador para generar datos de gráficos por

ordenador como el vídeo de origen, o una combinación de dichas fuentes. En un ejemplo, si la fuente de vídeo 18 es una videocámara, el dispositivo de origen 12 y el dispositivo de destino 14 pueden formar los denominados teléfonos con cámara o videoteléfonos. Sin embargo, las técnicas descritas en la presente divulgación pueden ser aplicables a la codificación de vídeo en general, y se pueden aplicar a aplicaciones inalámbricas y/o por cable.

[0032] El vídeo captado, precaptado o generado por ordenador se puede codificar por el codificador de vídeo 20. Los datos de vídeo codificados se pueden transmitir directamente al dispositivo de destino 14 por medio de la interfaz de salida 22 del dispositivo de origen 12. Los datos de vídeo codificados se pueden almacenar también (o de forma alternativa) en el dispositivo de almacenamiento 17 para un posterior acceso por el dispositivo de destino 14 u otros dispositivos, para su descodificación y/o reproducción.

[0033] El dispositivo de destino 14 incluye una interfaz de entrada 28, un descodificador de vídeo 30 y un dispositivo de visualización 32. En algunos casos, la interfaz de entrada 28 puede incluir un receptor y/o un módem. La interfaz de entrada 28 del dispositivo de destino 14 recibe los datos de vídeo codificados a través del enlace 16. Los datos de vídeo codificados comunicados sobre el enlace 16, o proporcionados en el dispositivo de almacenamiento 17, pueden incluir una variedad de elementos sintácticos generados por el codificador de vídeo 20 para su uso por un descodificador de vídeo, tal como el descodificador de vídeo 30, en la descodificación de los datos de vídeo. Dichos elementos sintácticos se pueden incluir con los datos de vídeo codificados, transmitidos en un medio de comunicación, almacenados en un medio de almacenamiento o almacenados en un servidor de archivos.

[0034] El dispositivo de visualización 32 se puede integrar con, o ser externo a, el dispositivo de destino 14. En algunos ejemplos, el dispositivo de destino 14 puede incluir un dispositivo de visualización integrado y también estar configurado para interconectarse con un dispositivo de visualización externo. En otros ejemplos, el dispositivo de destino 14 puede ser un dispositivo de visualización. En general, el dispositivo de visualización 32 visualiza los datos de vídeo descodificados ante un usuario, y puede comprender cualquiera de una variedad de dispositivos de visualización, tales como una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodos orgánicos emisores de luz (OLED) u otro tipo de dispositivo de visualización.

[0035] En algunos ejemplos, el codificador de vídeo 20 y el descodificador de vídeo 30 funcionan de acuerdo con un estándar de compresión de vídeo, tal como ISO/IEC MPEG-4 Visual e ITU-T H.264 (también conocido como ISO/IEC MPEG-4 AVC), incluida su extensión de codificación de vídeo escalable (SVC), extensión de codificación de vídeo de múltiples visualizaciones (MVC) y extensión 3DV basada en MVC. En algunos casos, cualquier flujo de bits que se ajuste a la extensión 3DV basada en MVC de H.264/AVC siempre contiene un subflujo de bits que es compatible con la extensión MVC de H.264/AVC. Además, actualmente se están emprendiendo unas iniciativas para generar una ampliación de codificación de vídeo tridimensional (3DV) para H.264/AVC, en concreto, 3DV basado en AVC. En otros ejemplos, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar de acuerdo con la Recomendación ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 o ISO/IEC MPEG-2 Visual e ITU-T H.264, ISO/IEC Visual.

[0036] Recientemente, el Equipo Mixto de Colaboración sobre Codificación de Vídeo (JCT-VC) del Grupo de Expertos en Codificación de Vídeo (VCEG) de ITU-T y el Grupo de Expertos en Imágenes en Movimiento (MPEG) de ISO/IEC ha terminado el diseño de un nuevo estándar de codificación de vídeo, concretamente la codificación de vídeo de alta eficacia (HEVC). El codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar de acuerdo con el estándar HEVC.

[0037] Además, actualmente se están dedicando esfuerzos continuos a la elaboración de codificación de vídeo escalable, codificación de múltiples visualizaciones y extensiones 3DV para HEVC. Dos extensiones de la HEVC con soporte de servicios tridimensionales han estado en desarrollo por parte del Equipo Conjunto de Colaboración en Codificación de Vídeo Tridimensional (JCT-3V) del VCEG de ITU-T y el MPEG de ISO/IEC y se denominan MV-HEVC y 3D-HEVC, respectivamente. El estándar 3D-HEVC codifica vídeo de múltiples vistas más el formato de profundidad e incluye nuevas herramientas de codificación integradas además de los módulos de codificación de la HEVC. Las herramientas de codificación recién introducidas son aplicables tanto para la codificación de texturas como para la codificación de profundidad. MV-HEVC se refiere a la extensión de codificación de múltiples vistas de HEVC, que da soporte a la codificación de múltiples vistas (de textura) sin cambiar el diseño de nivel de bloque de HEVC. Un reciente borrador de trabajo (WD) de MV-HEVC, denominado WD4 de aquí en adelante en el presente documento, está disponible desde el 25 de septiembre de 2014 en http://phenix.it-sudparis.eu/jct2/doc_end_user/documents/4_Incheon/wg11/JCT3V-D1004-v1.zip.

[0038] La extensión de codificación de vídeo escalable de HEVC puede denominarse SHEVC. Un borrador de trabajo (WD) reciente de SHEVC, y denominado SHEVC WD2 de aquí en adelante en el presente documento, está disponible desde el 25 de septiembre de 2014 en http://phenix.int-evry.fr/jct/doc_end_user/documents/13_Incheon/wg11/JCTVC-M1008-v1.zip. El codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar de acuerdo con cualquiera de SHEVC, MV-HEVC y/o 3D-HEVC.

[0039] Aunque no se muestra en la FIG. 1, en algunos aspectos, tanto el codificador de vídeo 20 como el descodificador de vídeo 30 pueden estar integrados con un codificador y descodificador de audio, y pueden incluir unidades MUX-DEMUX adecuadas, u otro tipo de hardware y software, para gestionar la codificación tanto de audio como de vídeo en un flujo de datos común o en flujos de datos separados. Si procede, en algunos ejemplos, las unidades MUX-DEMUX pueden ajustarse al protocolo de multiplexor ITU H.223 o a otros protocolos, tales como el protocolo de datagramas de usuario (UDP).

[0040] Tanto el codificador de vídeo 20 como el descodificador de vídeo 30 se pueden implementar como cualquiera de una variedad de circuitos codificadores adecuados, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados específicos de la aplicación (ASIC), matrices de puertas programables in situ (FPGA), lógica discreta, software, hardware, firmware o cualquier combinación de los mismos. Cuando las técnicas se implementan parcialmente en software, un dispositivo puede almacenar instrucciones para el software en un medio adecuado no transitorio legible por ordenador y ejecutar las instrucciones en hardware usando uno o más procesadores para realizar las técnicas de la presente divulgación. Tanto el codificador de vídeo 20 como el descodificador de vídeo 30 se pueden incluir en uno o más codificadores o descodificadores, de los que cualquiera se puede integrar como parte de un codificador/descodificador (CÓDEC) combinado en un dispositivo respectivo.

[0041] Como se indica anteriormente, el JCT-VC ha terminado recientemente de elaborar el estándar HEVC. Los trabajos de estandarización HEVC se basaron en un modelo en evolución de un dispositivo de codificación de vídeo denominado modelo de prueba HEVC (HM). El HM supone varias capacidades adicionales de los dispositivos de codificación de vídeo respecto a los dispositivos existentes, de acuerdo con, por ejemplo, la norma ITU-T H.264/AVC. Por ejemplo, mientras que la norma H.264 proporciona nueve modos de codificación mediante intrapredicción, el HM puede proporcionar hasta treinta y tres modos de codificación mediante intrapredicción.

[0042] En general, el modelo de trabajo del HM describe que una trama o imagen de vídeo se puede dividir en una secuencia de bloques arbolados o unidades de codificación más grandes (LCU), que incluyen tanto muestras de luminancia como de crominancia. Un bloque arbolado tiene un propósito similar al de un macrobloque de la norma H.264. Un fragmento incluye un número de bloques arbolados consecutivos en orden de codificación. Una trama o imagen de vídeo se puede dividir en uno o más fragmentos. Cada bloque arbolado se puede dividir en unidades de codificación (CU) de acuerdo con un árbol cuaternario. Por ejemplo, un bloque arbolado, como un nodo raíz del árbol cuaternario, se puede separar en cuatro nodos hijo, y cada nodo hijo puede, a su vez, ser un nodo padre y separarse en otros cuatro nodos hijo. Un nodo hijo final, no separado, como un nodo hoja del árbol cuaternario, comprende un nodo de codificación, es decir, un bloque de vídeo codificado. Los datos sintácticos asociados a un flujo de bits codificado pueden definir un número máximo de veces que se puede separar un bloque arbolado, y también pueden definir un tamaño mínimo de los nodos de codificación.

[0043] Una CU incluye un nodo de codificación y unidades de predicción (PU) y unidades de transformada (TU) asociadas al nodo de codificación. Un tamaño de la CU corresponde a un tamaño del nodo de codificación y debe tener conformación cuadrada. El tamaño de la CU puede variar desde 8x8 píxeles hasta el tamaño del bloque arbolado, con un máximo de 64x64 píxeles o más. Cada CU puede contener una o más PU y una o más TU. Los datos sintácticos asociados a una CU pueden describir, por ejemplo, la división de la CU en una o más PU. Los modos de división pueden diferir entre sí la CU está codificada en modo directo o de salto, codificada en modo de intrapredicción o codificada en modo de interpretación. Las PU se pueden dividir para tener una conformación no cuadrada. Los datos sintácticos asociados a una CU también pueden describir, por ejemplo, la división de la CU en una o más TU de acuerdo con un árbol cuaternario. Una TU puede tener una conformación cuadrada o no cuadrada.

[0044] La norma HEVC admite transformaciones de acuerdo con las TU, que pueden ser diferentes para CU diferentes. El tamaño de las TU se basa típicamente en el tamaño de las PU dentro de una CU dada definida para una LCU dividida, aunque puede que no sea siempre así. Las TU son típicamente del mismo tamaño o más pequeñas que las PU. En algunos ejemplos, las muestras residuales correspondientes a una CU se pueden subdividir en unidades más pequeñas usando una estructura de árbol cuaternario conocida como "árbol cuaternario residual" (RQT). Los nodos hoja del RQT pueden denominarse unidades de transformada (TU). Los valores de diferencias de píxeles asociados a las TU se pueden transformar para producir coeficientes de transformada, que se pueden cuantificar.

[0045] En general, una PU incluye datos relacionados con el procedimiento de predicción. Por ejemplo, cuando la PU se codifica en intramodo, la PU puede incluir datos que describen un modo de intrapredicción para la PU. Como otro ejemplo, cuando la PU se codifica en intermodo, la PU puede incluir datos que definen un vector de movimiento para la PU. Los datos que definen el vector de movimiento para una PU pueden describir, por ejemplo, una componente horizontal del vector de movimiento, una componente vertical del vector de movimiento, una resolución para el vector de movimiento (por ejemplo, precisión de un cuarto de píxel o precisión de un octavo de píxel), una imagen de referencia a la que apunta el vector de movimiento y/o una lista de imágenes de referencia (por ejemplo, lista 0, lista 1 o lista C) para el vector de movimiento.

[0046] Como se describirá con más detalle a continuación, como parte de la codificación entre modos, el codificador de vídeo 20 puede determinar si se debe usar el mismo tipo de dependencia para todas las capas de datos de vídeo multicapa o usar múltiples tipos de dependencia. El codificador de vídeo 20 puede señalar u omitir a continuación la señalización en base al tipo de dependencia usado. El descodificador de vídeo 30 está configurado con, o potencialmente recibe por medio de la señalización, un tipo predeterminado, de modo que si el tipo de dependencia no está señalizado, el descodificador de vídeo 30 puede simplemente aplicar el tipo predeterminado.

[0047] En general, se usa una TU para los procedimientos de transformada y cuantificación. Una CU dada que tiene una o más PU también puede incluir una o más unidades de transformada (TU). Seguidamente a la predicción, el codificador de vídeo 20 puede calcular valores residuales correspondientes a la PU. Los valores residuales comprenden valores de diferencias de píxeles que se pueden transformar en coeficientes de transformada, cuantificar y explorar usando las TU, para producir coeficientes de transformada en serie para la codificación por entropía. Esta divulgación usa típicamente el término "bloque de vídeo" para referirse a un nodo de codificación de una CU. En algunos casos específicos, la presente divulgación también puede usar el término "bloque de vídeo" para referirse a un bloque arbolado, es decir, una LCU o una CU, que incluye un nodo de codificación y PU y TU.

[0048] Una secuencia de vídeo incluye típicamente una serie de tramas o imágenes de vídeo. Un grupo de imágenes (GOP) comprende, en general, una serie de una o más de las imágenes de vídeo. Un GOP puede incluir datos sintácticos en una cabecera del GOP, una cabecera de una o más de las imágenes, o en otras ubicaciones, que describen una serie de imágenes incluidas en el GOP. Cada fragmento de una imagen puede incluir datos sintácticos de fragmento que describen un modo de codificación para el fragmento respectivo. El codificador de vídeo 20 funciona típicamente en bloques de vídeo dentro de fragmentos de vídeo individuales para codificar los datos de vídeo. Un bloque de vídeo puede corresponder a un nodo de codificación dentro de una CU. Los bloques de vídeo pueden tener tamaños fijos o variables y pueden diferir en tamaño de acuerdo con una norma de codificación especificada.

[0049] En un ejemplo, el HM admite la predicción en diversos tamaños de PU. Si se supone que el tamaño de una CU particular es $2N \times 2N$, el HM admite la intrapredicción en tamaños de PU de $2N \times 2N$ o $N \times N$, así como la interpredicción en tamaños de PU simétricas de $2N \times 2N$, $2N \times N$, $N \times 2N$ o $N \times N$. El HM también admite la división asimétrica para la interpredicción en tamaños de PU de $2N \times nU$, $2N \times nD$, $nL \times 2N$ y $nR \times 2N$. En la división asimétrica, una dirección de una CU no está dividida, mientras que la otra dirección está dividida entre un 25 % y un 75 %. La parte de la CU correspondiente a la división del 25 % está indicada por una "n" seguida de una indicación de "arriba", "abajo", "izquierda" o "derecha". Así pues, por ejemplo, " $2N \times nU$ " se refiere a una CU de tamaño $2N \times 2N$ que está dividida horizontalmente, con una PU de tamaño $2N \times 0,5N$ encima y una PU de tamaño $2N \times 1,5N$ debajo.

[0050] En la presente divulgación, " $N \times N$ " y " N por N " se pueden usar de manera intercambiable para hacer referencia a las dimensiones de píxeles de un bloque de vídeo en términos de dimensiones verticales y horizontales, por ejemplo, 16×16 píxeles o 16 por 16 píxeles. En general, un bloque de 16×16 tendrá 16 píxeles en una dirección vertical ($y = 16$) y 16 píxeles en una dirección horizontal ($x = 16$). Asimismo, un bloque de tamaño $N \times N$ tiene, en general, N píxeles en una dirección vertical y N píxeles en una dirección horizontal, donde N representa un valor entero no negativo. Los píxeles de un bloque se pueden disponer en filas *et alumnas*. Además, no es necesario que los bloques tengan necesariamente el mismo número de píxeles en la dirección horizontal que en la dirección vertical. Por ejemplo, los bloques pueden comprender $N \times M$ píxeles, donde M no es necesariamente igual a N .

[0051] Después de la codificación de intrapredicción o interpredicción mediante las PU de una CU, el codificador de vídeo 20 puede calcular datos residuales para las TU de la CU. Las PU pueden comprender datos de píxeles en el dominio espacial (también denominado dominio de píxel) y las TU pueden comprender coeficientes en el dominio de transformada seguidamente a la aplicación de una transformada, por ejemplo, una transformada discreta de coseno (DCT), una transformada de enteros, una transformada de ondículas o una transformada conceptualmente similar a los datos de vídeo residuales. Los datos residuales pueden corresponder a diferencias de píxeles entre píxeles de la imagen no codificada y valores de predicción correspondientes a las PU. El codificador de vídeo 20 puede formar las TU, incluyendo los datos residuales para la CU y, a continuación, transformar las TU para producir coeficientes de transformada para la CU.

[0052] Después de cualquier transformada para producir coeficientes de transformada, el codificador de vídeo 20 puede realizar la cuantificación de los coeficientes de transformada. La cuantificación se refiere, en general, a un procedimiento en el que coeficientes de transformada se cuantifican para reducir posiblemente la cantidad de datos usados para representar los coeficientes, proporcionando compresión adicional. El proceso de cuantificación puede reducir la profundidad de bits asociada con algunos, o la totalidad, de los coeficientes. Por ejemplo, un valor de n bits puede redondearse a la baja a un valor de m bits durante la cuantificación, donde n es mayor que m .

[0053] En algunos ejemplos, el codificador de vídeo 20 puede usar un orden de exploración predefinido para explorar los coeficientes de transformada cuantificados, para producir un vector en serie que se pueda codificar

por entropía. En otros ejemplos, el codificador de vídeo 20 puede realizar una exploración adaptativa. Después de explorar los coeficientes de transformada cuantificados para formar un vector unidimensional, el codificador de vídeo 20 puede realizar la codificación por entropía del vector unidimensional, por ejemplo, de acuerdo con la codificación de longitud variable adaptativa al contexto (CAVLC), la codificación aritmética binaria adaptativa al contexto (CABAC), la codificación aritmética binaria adaptativa al contexto basada en la sintaxis (SBAC), la codificación por entropía por división de intervalos de probabilidad (PIPE) u otra metodología de codificación por entropía. El codificador de vídeo 20 también puede codificar por entropía los elementos sintácticos asociados a los datos de vídeo codificados, para su uso por el descodificador de vídeo 30 en la descodificación de los datos de vídeo.

[0054] Para realizar la CABAC, el codificador de vídeo 20 puede asignar un contexto, dentro de un modelo de contexto, a un símbolo que se va a transmitir. El contexto se puede referir, por ejemplo, a si los valores vecinos del símbolo son distintos de cero o no. Para realizar la codificación CAVLC, el codificador de vídeo 20 puede seleccionar un código de longitud variable para un símbolo a transmitir. Las palabras de código en la VLC se pueden construir de modo que los códigos relativamente más cortos correspondan a símbolos más probables, mientras que los códigos más largos correspondan a símbolos menos probables. De esta forma, el uso de la VLC puede lograr un ahorro en bits con respecto, por ejemplo, al uso de palabras de código de igual longitud para cada símbolo que se va a transmitir. La determinación de la probabilidad se puede basar en un contexto asignado al símbolo.

[0055] Como se indica anteriormente, esta divulgación analiza técnicas relacionadas con la señalización del tipo de dependencia entre capas en la codificación de vídeo multicapa, incluyendo la MV-HEVC. Las técnicas descritas en esta divulgación también pueden aplicarse potencialmente a otras extensiones multicapa de HEVC, tal como SHEVC u otros códecs. Como se indica anteriormente, MVC es una extensión de H.264/AVC. Los aspectos de la especificación de MVC se analizarán ahora brevemente para proporcionar el contexto y los antecedentes para futuros análisis.

[0056] La FIG. 2 es un diagrama gráfico que ilustra un orden de codificación o descodificación MVC ejemplar, de acuerdo con uno o más ejemplos descritos en esta divulgación. Por ejemplo, la disposición de orden de descodificación ilustrada en la FIG. 2 se denomina codificación de tiempo primero. En la FIG. 2, S0 a S7 se refieren, cada uno, a diferentes vistas del vídeo multivista. Cada una entre T0 a T8 representa una instancia de tiempo de salida. Una unidad de acceso puede incluir las imágenes codificadas de todas las vistas para una instancia de tiempo de salida. Por ejemplo, una primera unidad de acceso incluye todas las vistas S0-S7 para la instancia de tiempo T0 (es decir, las imágenes 0-7), una segunda unidad de acceso incluye todas las vistas S0-S7 para la instancia de tiempo T1 (es decir, las imágenes 8-15), etc. En este ejemplo, las imágenes 0-7 están en una misma instancia de tiempo (es decir, la instancia de tiempo T0), las imágenes 8-15 en una misma instancia de tiempo (es decir, la instancia de tiempo T1). Las imágenes con la misma instancia de tiempo se muestran en general al mismo tiempo, y es la disparidad horizontal, y posiblemente alguna disparidad vertical, entre los objetos dentro de las imágenes de la misma instancia de tiempo las que hacen que el espectador perciba una imagen que abarca un volumen en 3D.

[0057] Se muestra un orden de descodificación MVC típico (es decir, un orden de flujo de bits) en la FIG. 2. La disposición del orden de descodificación se denomina codificación de tiempo en primer lugar. Cada unidad de acceso se define para contener las imágenes codificadas de todas las vistas para un instante de tiempo de salida. Se debe tener en cuenta que el orden de descodificación de las unidades de acceso puede no ser idéntico al orden de salida o de visualización.

[0058] La FIG. 3 es un diagrama conceptual que ilustra un patrón de predicción de la codificación de vídeo de múltiples vistas (MVC) de ejemplo. En el ejemplo de la FIG. 3, se ilustran ocho vistas y se ilustran doce ubicaciones temporales para cada visualización. En general, cada fila en la FIG. 3 corresponde a una vista, mientras que cada columna indica una ubicación temporal. Cada una de las vistas se puede identificar usando un identificador de vista ("view_id"), que se puede usar para indicar una ubicación de cámara relativa con respecto a las otras vistas. En el ejemplo mostrado en la FIG. 3, los ID de vista se indican como de "S0" a "S7", aunque también se pueden usar ID de vista numéricos. Además, cada una de las ubicaciones temporales se puede identificar usando un valor de recuento de orden de imágenes (POC), que indica un orden de visualización de las imágenes. En el ejemplo mostrado en la FIG. 3, los valores de POC se indican como de "T0" a "T11".

[0059] Las imágenes de la FIG. 3 se indican usando un bloque sombreado que incluye una letra, que designa si la imagen correspondiente está intracodificada (es decir, una trama I), o intercodificada en una dirección (es decir, como una trama P) o en múltiples direcciones (es decir, como una trama B). En general, las predicciones se indican mediante flechas, donde la imagen a la que se apunta usa el objeto desde el que se apunta como referencia de predicción. Por ejemplo, la trama P de la vista S2 en la ubicación temporal T0 se predice a partir de la trama I de la vista S0 en la ubicación temporal T0. Cada una de las imágenes mostradas en la FIG. 3 se puede denominar un componente de vista.

5 **[0060]** Al igual que con la codificación de vídeo de vista única, las imágenes de una secuencia de vídeo de múltiples vistas pueden codificarse predictivamente con respecto a las imágenes en diferentes ubicaciones temporales. Por ejemplo, la trama b de la vista S0 en la ubicación temporal T1 tiene una flecha apuntando a la misma desde la trama l de la vista S0 en la ubicación temporal T0, que indica que la trama b se predice a partir de la trama l. Adicionalmente, sin embargo, en el contexto de la codificación de vídeo multivista, las imágenes se pueden predecir entre vistas. Es decir, un componente de la vista puede usar los componentes de la vista en otras vistas como referencia. En la MVC, por ejemplo, la predicción entre vistas se realiza como si el componente de la vista de otra vista fuera una referencia de interpredicción. Las posibles referencias entre vistas se pueden señalar en la extensión SPS MVC y se pueden modificar por el proceso de construcción de la lista de imágenes de referencia, que permite el ordenamiento flexible de las referencias de interpredicción o de predicción entre vistas.

15 **[0061]** La FIG. 3 proporciona diversos ejemplos de predicción entre vistas. Las imágenes de la vista S1, en el ejemplo de la FIG. 3, se ilustran como que se predicen a partir de imágenes en diferentes ubicaciones temporales de la vista S1, así como entre vistas predichas a partir de imágenes de las imágenes de vistas S0 y S2 en las mismas ubicaciones temporales. Por ejemplo, la trama b de la vista S1 en la ubicación temporal T1 se predice a partir de cada una de las tramas B de la vista S1 en las ubicaciones temporales T0 y T2, así como las tramas b de las vistas S0 y S2 en la ubicación temporal T1.

20 **[0062]** En el ejemplo de la figura 3, la "B" mayúscula y la "b" minúscula están destinadas a indicar diferentes relaciones jerárquicas entre imágenes, en lugar de diferentes metodologías de codificación. En general, las tramas con "B" mayúscula están relativamente más altas en la jerarquía de predicción que las tramas con "b" minúscula. La figura 3 también ilustra variaciones en la jerarquía de predicción usando diferentes niveles de sombreado, donde las imágenes con una mayor cantidad de sombreado (es decir, relativamente más oscuras) están más altas en la jerarquía de predicción que aquellas imágenes que tienen menos sombreado (es decir, relativamente más claras). Por ejemplo, todas las tramas l en la figura 3 se ilustran con sombreado completo, mientras que las tramas P tienen un sombreado algo más claro, y las tramas B (y las tramas b minúscula) tienen diversos niveles de sombreado entre sí, pero siempre más claro que el sombreado de las tramas P y las tramas l.

30 **[0063]** En general, la jerarquía de predicción se relaciona con índices del orden de visualización, en cuanto a que las imágenes relativamente más altas en la jerarquía de predicción deberían ser decodificadas antes de la descodificación de imágenes que están relativamente más bajas en la jerarquía, de tal modo que aquellas imágenes relativamente más altas en la jerarquía se puedan usar como imágenes de referencia durante la descodificación de las imágenes relativamente más bajas en la jerarquía. Un índice de orden de vista es un índice que indica el orden de descodificación de componentes de vista en una unidad de acceso. Los índices de orden de vista pueden estar implícitos en un conjunto de parámetros, tal como un SPS.

40 **[0064]** De esta manera, las imágenes utilizadas como imágenes de referencia pueden ser decodificadas antes de la descodificación de las imágenes que se codifican con referencia a las imágenes de referencia. Un índice de orden de vista es un índice que indica el orden de descodificación de componentes de vista en una unidad de acceso. De acuerdo con MVC/AVC, para cada índice i de orden de vista, se señala el id de vista correspondiente. La descodificación de los componentes de vista sigue el orden ascendente de los índices de orden de vista. Si se presentan todas las vistas, entonces el conjunto de índices de orden de vista comprende un conjunto consecutivamente ordenado de cero a uno menos que el número total de vistas.

45 **[0065]** En algunos casos, un subconjunto de todo un flujo de bits puede ser extraído para formar un subflujo de bits que todavía se ajusta a la MVC. Existen muchos posibles subflujos de bits que las aplicaciones específicas pueden requerir, basándose, por ejemplo, en un servicio proporcionado por un servidor, la capacidad, el soporte y las capacidades de los descodificadores de uno o más clientes y/o la preferencia de uno o más clientes. Por ejemplo, un cliente podría requerir solo tres vistas, y podría haber dos entornos. En un ejemplo, un cliente puede requerir una experiencia visual sin fisuras y podría preferir vistas con valores de view_id S0, S1 y S2, mientras que otro cliente puede requerir ajustabilidad a escala de las vistas y preferir vistas con valores de view_id S0, S2 y S4. Ambos de estos subflujos de bits se pueden descodificar como flujos de bits de MVC independientes y se pueden admitir simultáneamente.

55 **[0066]** Mientras que la FIG. 3 muestra ocho vistas (S0-S7), como se indica anteriormente, la extensión MVC/AVC soporta hasta 1024 vistas y usa un ID de vista en una cabecera de unidad de NAL para identificar la vista a la que pertenece una unidad de NAL. Un índice de orden de vista se puede señalar como parte de una cabecera de unidad NAL. Es decir, para propósitos de comparación, un índice de orden de vista puede reemplazar el view_id que se señala en la cabecera de unidad NAL de la extensión MVC/AVC. El orden de vista en general describe el ordenamiento de las vistas en una unidad de acceso, y un índice de orden de vista identifica una vista particular en el orden de vista de la unidad de acceso. Es decir, un índice de orden de vista describe el orden de descodificación de un componente de vista correspondiente de una unidad de acceso.

65 **[0067]** En consecuencia, un SPS puede proporcionar una relación entre view_ids para las vistas y los índices de orden de vista para las vistas. Usando el índice de orden de vista y los datos en el SPS, el codificador de vídeo 20

y el descodificador de vídeo 30 pueden reemplazar el id de vista de 10 bits de MVC/AVC en la cabecera de unidad NAL por el índice de orden de vista, que puede dar lugar a un ahorro de bits con respecto al esquema MVC/AVC.

5 **[0068]** Una estructura de predicción MVC típica (que incluye tanto la predicción entre imágenes dentro de cada vista como la predicción entre vistas) para la codificación de vídeo multivista se muestra en la FIG. 3. Las predicciones se indican con flechas. En el ejemplo de la FIG. 3, el objeto que apunta usa el objeto apuntado para referencia de predicción.

10 **[0069]** En MVC, la predicción entre vistas se soporta mediante la compensación de movimiento de disparidad, que utiliza la sintaxis de la compensación de movimiento H.264/AVC, pero permite usar una imagen de una vista diferente como imagen de referencia. La codificación de dos vistas también puede ser soportada por la MVC. Una de las ventajas potenciales de la MVC es que un codificador de MVC podría tomar más de dos vistas como una entrada de vídeo 3D y un descodificador de MVC puede descodificar dicha representación de múltiples vistas. Así, cualquier renderizador con un descodificador de MVC puede esperar contenidos de vídeo 3D con más de dos vistas.

15 **[0070]** Ahora se describirán aspectos de la predicción entre vistas de MVC. En la MVC, se permite la predicción entre vistas entre imágenes en la misma unidad de acceso (es decir, con la misma instancia de tiempo). Cuando se codifica una imagen en una de las vistas no básicas, se puede añadir una imagen a una lista de imágenes de referencia si está en una vista diferente pero con una misma instancia de tiempo. Una imagen de referencia de predicción entre vistas se puede poner en cualquier posición de una lista de imágenes de referencia, tal como cualquier imagen de referencia de interpredicción.

20 **[0071]** Ahora se describirán los aspectos del conjunto de parámetros de secuencia (SPS) para la extensión de MVC. Como se muestra en la FIG. 3, un componente de la vista puede usar los componentes de la vista en otras vistas como referencia, lo cual se conoce como predicción entre vistas. En la MVC, la predicción entre vistas se realiza como si el componente de la vista en otra visualización fuera una referencia de interpredicción. Sin embargo, las posibles referencias entre vistas se señalizan en la extensión de la MVC del SPS (como se muestra en la TABLA 1 a continuación) y pueden ser modificadas por el proceso de construcción de la lista de imágenes de referencia implementada por un codificador de vídeo, que habilita el ordenamiento flexible de las referencias entre predicciones o de predicción entre vistas.

TABLA 1

	Descriptor
seq_parameter_set_mvc_extension() {	
num_views_minus1	ue(v)
for(i = 0; i <= num_views_minus1; i++)	
view_id[i]	ue(v)
for(i = 1; i <= num_views_minus1; i++){	
num_anchor_refs_I0[i]	ue(v)
for(j = 0; j < num_anchor_refs_I0[i]; j++)	
anchor_ref_I0[i][j]	ue(v)
num_anchor_refs_I1[i]	ue(v)
for(j = 0; j < num_anchor_refs_I1[i]; j++)	
anchor_ref_I1[i][j]	ue(v)
}	
for(i = 1; i <= num_views_minus1; i++){	
num_non_anchor_refs_I0[i]	ue(v)
for(j = 0; j < num_non_anchor_refs_I0[i]; j++)	
non_anchor_ref_I0[i][j]	ue(v)
num_non_anchor_refs_I1[i]	ue(v)
for(j = 0; j < num_non_anchor_refs_I1[i]; j++)	
non_anchor_ref_I1 [i][j]	ue(v)
}	
num_level_values_signalled_minus1	ue(v)
for(i = 0; i <= num_level_values_signalled_minus1; i++){	
level_idc[i]	u(8)
num_applicable_ops_minus1[i]	ue(v)
for(j = 0; j <= num_applicable_ops_minus1[i]; j++){	
applicable_op_temporal_id[i][j]	u(3)
applicable_op_num_target_views_minus1[i][j]	ue(v)
for(k = 0; k <= applicable_op_num_target_views_minus1[i][j]; k++)	

applicable_op_target_view_id [i][j][k]	ue(v)
applicable_op_num_views_minus1 [i][j]	ue(v)
}	
}	
}	

[0072] En la extensión de MVC del SPS mostrada anteriormente en la TABLA 1, para cada vista, se señala el número de vistas que se pueden usar para formar la lista de imágenes de referencia 0 y la lista de imágenes de referencia 1.

5

[0073] La relación de predicción para una imagen de anclaje, tal como se señala en la extensión de MVC del SPS, puede ser diferente de la relación de predicción para una imagen sin anclaje (señalada en la extensión de MVC del SPS) de la misma vista. En H.264, una imagen de anclaje es una imagen codificada en la que todos los fragmentos solo pueden hacer referencia a los fragmentos dentro de la misma unidad de acceso. En otras palabras, la predicción entre vistas se puede usar en una imagen de anclaje, pero no se usa interpretación (es decir, predicción temporal) para una imagen de anclaje. H.264 define una imagen sin anclaje como una imagen que no es una imagen de anclaje.

10

[0074] Ahora se analizarán aspectos del tipo de dependencia en MV-HEVC. En MV-HEVC, el tipo de dependencia entre capas se señala de manera que se da soporte a al menos tres tipos. Los tres tipos son predicción de muestra entre capas (como en MVC), predicción de movimiento entre capas y la combinación de ambas.

15

[0075] Como se muestra a continuación, en la extensión del conjunto de parámetros de vídeo (VPS) de la MV-HEVC, el elemento sintáctico "direct_dep_type_len_minus2" especifica cuántos tipos de predicción entre capas son compatibles, mientras que el elemento sintáctico "direct_dependency_type[i][j]" especifica el tipo exacto de predicción entre capas. Más específicamente, el elemento sintáctico "direct_dependency_type[i][j]" identifica el tipo de dependencia que la capa [i] tiene con respecto a la capa [j].

20

[0076] A continuación se reproduce una parte de MV-HEVC WD5 relacionada con la semántica de VPS.

25

Sintaxis de la extensión del conjunto de parámetros de vídeo

[0077]

30

TABLA 2

vps_extension() {	Descriptor
while(!byte_aligned())	
vps_extension_byte_alignment_reserved_one_bit	u(1)
avc_base_layer_flag	u(1)
splitting_flag	u(1)
for(i = 0, NumScalabilityTypes = 0; i < 16; i++){	
scalability_mask [i]	u(1)
NumScalabilityTypes += scalability_mask[i]	
}	
...	
num_output_layer_sets	ue(v)
for(i = 0; i < num output layer sets; i++){	
output_layer_set_idx_minus1 [i]	ue(v)
lsidx = output_layer_set_idx_minus1[i] + 1	
for(j = 0; j <= vps_max_layer_id; j++){	
if(layer_id_included_flag[lsidx][j])	
output_layer_flag [lsidx][j]	u(1)
}	
...	
direct_dep_type_len_minus2	ue(v)
for(i = 1; i <= vps_max_layers_minus1; i++){	
for(j = 0; j < i; j++){	
if(direct_dependency_flag[i][j])	
direct_dependency_type [i][j]	u(v)
}	

[0078] `direct_dep_type_len_minus2` plus 2 especifica el número de bits del elemento sintáctico `direct_dependency_type[i][j]`. En los flujos de bits conformes a la presente versión de esta especificación, el valor de `direct_dep_type_len_minus2` será igual a 0. Aunque el valor de `direct_dep_type_len_minus2` será igual a 0 en esta versión de esta especificación, los descodificadores permitirán que aparezcan otros valores de `direct_dep_type_len_minus2` en el intervalo de 0 a 30, inclusive, en la sintaxis.

[0079] `direct_dependency_type[i][j]` indica el tipo de dependencia entre la capa con `nuh_layer_id` igual a `layer_id_in_nuh[i]` y la capa con `nuh_layer_id` igual a `layer_id_in_nuh[j]`. `direct_dependency_type[i][j]` igual a 0 indica que la capa con id de capa `nuh` igual a `layer_id_in_nuh[j]` se usa tanto para predicción de muestra entre capas como para predicción de movimiento entre capas con id de placa `nuh` igual a `layer_id_in_nuh[i]`. `direct_dependency_type[i][j]` igual a 1 indica que la capa con id de capa `nuh` igual a `layer_id_in_nuh[j]` se usa para predicción de muestra entre capas, pero no para predicción de movimiento entre capas de la capa con id de capa `nuh` igual a `layer_id_in_nuh[i]`. `direct_dependency_type[i][j]` igual a 2 indica que la capa con id de capa `nuh` igual a `layer_id_in_nuh[j]` se usa para predicción de movimiento entre capas, pero no para predicción de muestra entre capas de la capa con id de capa `nuh` igual a `layer_id_in_nuh[i]`. Aunque el valor de `direct_dependency_type[i][j]` estará en el intervalo de 0 a 2, ambos inclusive, en esta versión de esta especificación, los descodificadores permitirán que aparezcan valores de `direct_dependency_type[i][j]` en el intervalo de 3 a 232 - 2, ambos inclusive, en la sintaxis.

[0080] Al codificar determinados tipos de vídeo multicapa, el rendimiento del codificador de vídeo 20 y el descodificador de vídeo 30 puede no potenciarse, o solo puede potenciarse mínimamente, por determinados tipos de predicción entre capas, lo que significa que el tipo específico de predicción entre capas no produce un intercambio deseable del índice de distorsión. Como un ejemplo, en algunos casos, el vídeo codificado de MV-HEVC puede no beneficiarse, o solo beneficiarse mínimamente, de la predicción de movimiento entre capas. De acuerdo con las técnicas de esta divulgación, el codificador de vídeo 20 puede hacer una determinación de que la predicción de movimiento entre capas debe desactivarse y que el descodificador de vídeo 30 puede asumir un tipo de predicción entre capas. Incluso si un codificador de vídeo tomara dicha determinación, de acuerdo con las técnicas de señalización existentes, el codificador de vídeo todavía puede necesitar señalar un elemento sintáctico que identifique un tipo de dependencia (por ejemplo, "direct_dependency_type") para cada combinación de capas en un flujo de bits, costando cada aparición del elemento sintáctico de tipo de dependencia al menos dos bits. Las técnicas existentes de MV-HEVC no incluyen ningún mecanismo para evitar esta señalización del elemento sintáctico "direct_dependency_type" aunque el elemento sintáctico puede no ser útil si, por ejemplo, la predicción de movimiento entre capas está deshabilitada.

[0081] Esta divulgación introduce técnicas para modificar la señalización de un tipo de dependencia para una capa actual en relación con una capa de referencia. En MV-HEVC, por ejemplo, el elemento sintáctico "direct_dep_type_len_minus2" puede modificarse para permitir que el número de tipos de dependencia sea igual a 0, lo que indica que un elemento sintáctico de tipo de dependencia no se señalará. Cuando el elemento sintáctico de tipo de dependencia no está señalado, el descodificador de vídeo 30 puede determinar que el tipo de dependencia de una capa actual en relación con una capa de referencia es un tipo predeterminado y, por lo tanto, proceder a descodificar un bloque de la capa actual usando la predicción entre capas conforme al tipo predeterminado. Como se analiza anteriormente, el tipo predeterminado puede definirse en el códec de vídeo o puede señalizarse como parte del primer elemento sintáctico (por ejemplo, `direct_dep_type_len_minus2`).

[0082] Si se implementan las técnicas de esta divulgación junto con MV-HEVC, por ejemplo, la señalización del elemento sintáctico "direct_dependency_type" puede evitarse dependiendo del valor de `direct_dep_type_len_minus2`. Si `direct_dep_type_len_minus2` es igual a un primer valor, entonces el descodificador de vídeo 30 puede, por ejemplo, determinar que el tipo de dependencia de una capa actual en relación con una capa de referencia es un tipo predeterminado sin recibir instancias del elemento sintáctico `direct_dependency_type`. En la medida en que el descodificador de vídeo 30 necesite (es decir, use) un valor para el elemento sintáctico "direct_dependency_type" para los procesos de descodificación, el descodificador de vídeo 30 puede establecer el valor para `direct_dependency_type` en el valor correspondiente al tipo predeterminado. Si `direct_dep_type_len_minus2` (es decir, el primer elemento sintáctico) es igual a un segundo valor, entonces el descodificador de vídeo 30 puede, por ejemplo, determinar un tipo de dependencia para una capa actual en relación con una capa de referencia al recibir una instancia del elemento sintáctico `direct_dependency_type`.

[0083] De acuerdo con las técnicas de esta divulgación, el 0-ésimo tipo de dependencia puede hacerse igual (o asociarse con) un tipo predeterminado de predicción entre capas. Por tanto, el descodificador de vídeo 30 puede, en base a un valor de cero para `direct_dep_type_len_minus2`, determinar que el tipo de dependencia para las capas asociadas con el VPS es el tipo predeterminado. El tipo predeterminado puede ser, por ejemplo, cualquiera de (1) predicción de muestra entre capas, (2) predicción de movimiento entre capas o (3) una combinación de predicción de muestra entre capas y predicción de movimiento entre capas.

[0084] De acuerdo con las técnicas de esta divulgación, la semántica de extensión de VPS puede cambiarse de la manera que se muestra a continuación. En la siguiente descripción, lo subrayado está destinado a mostrar adiciones, y está destinado a mostrar eliminaciones. Los cambios que se muestran a continuación se muestran en

relación con MV-HEVC WD5. En los ejemplos a continuación, el valor de `direct_dep_type_len` puede indicar que se usa un tipo de predicción fija para toda la secuencia de vídeo codificada. Por tanto, los elementos sintácticos `direct_dependency_type` pueden no estar presentes necesariamente en algunos ejemplos. Otra señalización, como se analiza anteriormente, también se puede implementar de modo que los elementos sintácticos `direct_dependency_type` no se señalicen y se deriven como un valor predeterminado.

Sintaxis de la extensión del conjunto de parámetros de vídeo

[0085]

TABLA 3

vps_extension() {	Descriptor
...	
direct_dep_type_len	ue(v)
for (i = 1; i <= vps_max_layers_minus1; i++)	
for (j = 0; j < i; j++)	
if (direct_dependency_flag[i][j] && direct_dep_type_len)	
direct_dependency_type [i][j]	u(v)
}	

[0086] En el ejemplo de la TABLA 3 anterior, el elemento sintáctico "`direct_dep_type_len_minus2`" se cambia a "`direct_dep_type_len`", lo que significa que el valor puede ser igual a cero. Cuando el descodificador de vídeo 30 recibe, en un VPS, un valor de cero para "`direct_dep_type_len`", el descodificador de vídeo 30 asigna un tipo predeterminado de predicción entre capas a todas las combinaciones de capas asociadas con el VPS. Este tipo predeterminado puede, por ejemplo, programarse en un códec o puede señalizarse por separado. En el ejemplo de la TABLA 3 anterior, así como en otros ejemplos a continuación, si el descodificador de vídeo 30 recibe el elemento sintáctico `direct_dependency_type` en el flujo de bits, entonces el descodificador de vídeo 30 recibe una instancia de `direct_dependency_type` para cada combinación de capas, donde [i] y [j] en la TABLA 3 representan las diversas capas y [i][j] corresponde a una combinación de capas. Cuando el valor de "`direct_dep_type_len`" es cero, se omite el bucle mostrado anteriormente en la TABLA 3 (es decir, no se realiza el conjunto de instrucciones del bucle) porque la operación AND lógica "`direct_dependency_flag[i][j] && direct_dep_type_len`" es falsa porque `direct_dep_type_len` es falso. Por tanto, cuando se omite el bucle, el descodificador de vídeo 30 no recibe ninguna instancia del elemento sintáctico "`direct_dependency_type[i][j]`".

[0087] De acuerdo con las técnicas de esta divulgación, el codificador de vídeo 20 puede codificar datos de vídeo de una manera que el descodificador de vídeo 30 puede evitar (es decir, abstenerse y/o no necesitar) descodificar múltiples instancias del elemento sintáctico `direct_dependency_type` descodificando una instancia del elemento sintáctico `direct_dep_type_len`, reduciendo de este modo las sobrecargas de señalización. Sin embargo, las técnicas de esta divulgación todavía permiten que el codificador de vídeo 20 señale por separado un `direct_dependency_type` para cada combinación de capas cuando dicha señalización separada produce una calidad de codificación mejorada.

[0088] Aunque el elemento sintáctico "`direct_dep_type_len`" se muestra en la TABLA 3 anterior como un elemento sintáctico de longitud variable con un descriptor de "ue (v)", el elemento sintáctico "`direct_dep_type_len`" se puede señalar de forma alternativa usando codificación de longitud fija. Por ejemplo, el descriptor para el elemento sintáctico "`direct_dep_type_len`" en la TABLA 3 anterior se puede cambiar a u(5), u(6) o a cualquier otra longitud fija.

[0089] Adicionalmente, la condición del elemento sintáctico "`direct_dep_type_len`" que no es igual a 0 se puede colocar en un lugar anterior en la sintaxis de extensión del VPS, como se muestra en los ejemplos de la TABLA 4 y la TABLA 5 a continuación. Los ejemplos de la TABLA 4 y la TABLA 5 son implementaciones alternativas semánticamente pero son funcionalmente equivalentes. En los ejemplos de la TABLA 4 y la TABLA 5, un valor falso para el elemento sintáctico "`direct_dep_type_len`" termina el bucle usado para recibir las instancias de "`direct_dependency_type [i][j]`".

TABLA 4

vps_extension() {	Descriptor
...	
direct_dep_type_len	ue(v)
for (i = 1; i <= vps_max_layers_minus1 && direct_dep_type_len; i++)	
for (j = 0; j < i; j++)	
if (direct_dependency_flag[i][j])	

direct_dependency_type[i][j]	u(v)
}	

TABLA 5

	Descriptor
vps_extension() {	
...	
direct_dep_type_len	ue(v)
if(direct_dep_type_len)	
for(i = 1; i <= vps_max_layers_minus1; i++)	
for(j = 0; j < i; j++)	
if(direct_dependency_flag[i][j])	
direct_dependency_type[i][j]	u(v)
}	

5 **[0090]** Las modificaciones a las descripciones del elemento sintáctico "direct_dep_type_len_minus2" y el elemento sintáctico "direct_dependency_type[i][j]" se pueden implementar como sigue:

direct_dep_type_len especifica el número de bits del

10 elemento sintáctico direct_dependency_type[i][j]. El valor de direct_dep_type_len estará en el intervalo de 0 y 32, ambos inclusive.

NOTA - En la especificación de MV-HEVC, se puede requerir que el valor de direct_dep_type_len sea igual a 0. En la especificación de SHEVC, se puede requerir que el valor de direct_dep_type_len sea igual a 2.

15 direct_dependency_type[i][j] indica el tipo de dependencia entre la capa con nuh_layer_id igual a layer_id_in_nuh[i] y la capa con nuh_layer_id igual a layer_id_in_nuh[j]. direct_dependency_type[i][j] igual a 0 indica que la capa con id de capa nuh igual a layer_id_in_nuh[j] se usa tanto para predicción de muestra entre capas como para predicción de movimiento entre capas con id de placa nuh igual a layer_id_in_nuh[i]. direct_dependency_type[i][j] igual a 1 indica que la capa con id de capa nuh igual a layer_id_in_nuh[j] se usa para predicción de muestra entre capas, pero no para predicción de movimiento entre capas de la capa con id de capa nuh igual a layer_id_in_nuh[i]. direct_dependency_type[i][j] igual a 2 indica que la capa con id de capa nuh igual a layer_id_in_nuh[j] se usa para predicción de movimiento entre capas, pero no para predicción de muestra entre capas de la capa con id de capa nuh igual a layer_id_in_nuh[i]. Aunque el valor de direct_dependency_type[i][j] estará en el intervalo de 0 a 2, ambos inclusive, en esta versión de esta especificación, los decodificadores permitirán que aparezcan valores de direct_dependency_type[i][j] en el intervalo de 3 a $2^{32} - 2$, ambos inclusive, en la sintaxis. Quando direct_dependency_flag[i][j] es 1 y direct_dependency_type[i][j] no está presente, el valor de direct_dependency_type[i][j] se infiere que es igual a 1.

30 **[0091]** De forma alternativa, la semántica de direct_dependency_type[i][j] puede cambiarse para que el valor 0 corresponda solo al caso de predicción de muestra entre capas. En dicho ejemplo, la semántica puede cambiar como sigue.

35 direct_dependency_type[i][j] indica el tipo de dependencia entre la capa con nuh_layer_id igual a layer_id_in_nuh[i] y la capa con id de capa de nuh igual a layer_id_in_nuh[j]. direct_dependency_type[i][j] igual a 1 indica que la capa con id de capa nuh igual a layer_id_in_nuh[j] se usa tanto para predicción de muestra entre capas como para predicción de movimiento entre capas con id de placa nuh igual a layer_id_in_nuh[i]. direct_dependency_type[i][j] igual a 0 indica que la capa con id de capa nuh igual a layer_id_in_nuh[j] se usa para predicción de muestra entre capas, pero no para predicción de movimiento entre capas de la capa con id de capa nuh igual a layer_id_in_nuh[i]. direct_dependency_type[i][j] igual a 2 indica que la capa con nuh_layer_id igual a layer_id_in_nuh[j] se usa para predicción de movimiento entre capas, pero no para predicción de muestra entre capas de la capa con id de capa nuh igual a layer_id_in_nuh[i]. Aunque el valor de direct_dependency_type[i][j] estará en el intervalo de 0 a 2, ambos inclusive, en esta versión de esta especificación, los decodificadores permitirán que aparezcan valores de direct_dependency_type[i][j] en el intervalo de 3 a $2^{32} - 2$, ambos inclusive, en la sintaxis. Quando direct_dependency_flag[i][j] es igual a 1 y direct_dependency_type[i][j] no está presente, el valor de direct_dependency_type[i][j] se infiere que es igual a 0.

Cabe señalar que la posición de la oración comienza con "direct_dependency_type[i][j] igual a 1" y que la oración que comienza con "direct_dependency_type[i][j] igual a 0" puede intercambiarse.

50 **[0092]** En los ejemplos anteriores, inferir que el valor de "direct_dependency_type[i][j]" sea igual a 0 o 1 constituye un ejemplo de un tipo predeterminado de predicción entre capas.

[0093] La FIG. 4 es un diagrama de bloques que ilustra un codificador de vídeo 20 de ejemplo que puede implementar las técnicas descritas en esta divulgación. El codificador de vídeo 20 puede, por ejemplo, configurarse para generar vídeo codificado que implemente la semántica descrita anteriormente. En algunos casos, el codificador de vídeo 20 puede estar configurado para emitir vídeo a una entidad de posprocesamiento 27. La entidad de posprocesamiento 27 está destinada a representar un ejemplo de una entidad de vídeo, tal como un dispositivo MANE o de empalme/edición, que puede procesar datos de vídeo codificados del codificador de vídeo 20. En algunos casos, la entidad de posprocesamiento 27 puede ser un ejemplo de una entidad de red. En algunos sistemas de codificación de vídeo, la entidad de posprocesamiento 27 y el codificador de vídeo 20 pueden ser partes de dispositivos separados, mientras que, en otros casos, la funcionalidad descrita con respecto a la entidad de posprocesamiento 27 se puede realizar por el mismo dispositivo que comprende el codificador de vídeo 20.

[0094] El codificador de vídeo 20 puede realizar la intracodificación y la intercodificación de bloques de vídeo dentro de fragmentos de vídeo. La intracodificación se basa en la predicción espacial para reducir o eliminar la redundancia espacial de un vídeo dentro de una trama o imagen de vídeo dada. La intercodificación se basa en la predicción temporal para reducir o eliminar la redundancia temporal de un vídeo dentro de tramas o imágenes adyacentes de una secuencia de vídeo. El intramodo (modo I) se puede referir a cualquiera de varios modos de compresión de base espacial. Los intermodos, tales como la predicción unidireccional (modo P) o la bipredicción (modo B), se pueden referir a cualquiera de varios modos de compresión de base temporal.

[0095] En el ejemplo de la FIG. 4, el codificador de vídeo 20 incluye una memoria de datos de vídeo 31, una unidad de división 35, una unidad de procesamiento de predicción 41, una unidad de filtro 63, una memoria intermedia de imagen descodificada 64, un sumador 50, una unidad de procesamiento de transformada 52, una unidad de cuantización 54 y una unidad de codificación por entropía 56. La unidad de procesamiento de predicción 41 incluye una unidad de estimación de movimiento 42, una unidad de compensación de movimiento 44 y una unidad de procesamiento de intrapredicción 46. Para la reconstrucción de bloques de vídeo, el codificador de vídeo 20 incluye también una unidad de cuantificación inversa 58, una unidad de procesamiento de transformada inversa 60 y un sumador 62. La unidad de filtro 63 está destinada a representar uno o más filtros de bucle tales como un filtro de reducción del efecto bloque, un filtro de bucle adaptativo (ALF) y un filtro de desfase adaptativo de muestras (SAO). Aunque la unidad de filtro 63 que se muestra en la FIG. 2 es un filtro de bucle, en otras configuraciones la unidad de filtro 63 puede implementarse como un filtro postbucle.

[0096] La memoria de datos de vídeo 31 puede almacenar datos de vídeo a codificar mediante los componentes del codificador de vídeo 20. Los datos de vídeo almacenados en la memoria de datos de vídeo 31 se pueden obtener, por ejemplo, de la fuente de vídeo 18. La memoria intermedia de imágenes descodificadas 64 puede ser una memoria de imágenes de referencia que almacena datos de vídeo de referencia para su uso en la codificación de datos de vídeo mediante el codificador de vídeo 20, por ejemplo, en los modos de intracodificación o intercodificación. La memoria de datos de vídeo 31 y la memoria intermedia de imágenes descodificadas 64 pueden estar formadas por cualquiera de una variedad de dispositivos de memoria, tales como memoria dinámica de acceso aleatorio (DRAM), incluyendo DRAM síncrona (SDRAM), RAM magnetorresistiva (MRAM), RAM resistiva (RRAM) u otros tipos de dispositivos de memoria. El mismo dispositivo de memoria u otros dispositivos de memoria separados pueden proporcionar una memoria de datos de vídeo 31 y una memoria intermedia de imágenes descodificadas 64. En diversos ejemplos, la memoria de datos de vídeo 31 puede estar en un chip con otros componentes del codificador de vídeo 20, o fuera del chip en relación con esos componentes.

[0097] Como se muestra en la FIG. 4, el codificador de vídeo 20 recibe datos de vídeo en la memoria de datos de vídeo 31, y la unidad de división 35 divide los datos en bloques de vídeo. Esta partición también puede incluir la partición en fragmentos, mosaicos u otras unidades mayores, así como la división en bloques de vídeo, por ejemplo, de acuerdo con una estructura de árbol cuaternario de LCU y CU. El codificador de vídeo 20 ilustra, en general, los componentes que codifican bloques de vídeo dentro de un fragmento de vídeo que se va a codificar. El fragmento se puede dividir en múltiples bloques de vídeo (y, posiblemente, en conjuntos de bloques de vídeo denominados mosaicos). La unidad de procesamiento de predicción 41 puede seleccionar uno de una pluralidad de posibles modos de codificación, tal como uno de una pluralidad de modos de intracodificación, o uno de una pluralidad de modos de intercodificación, para el bloque de vídeo actual en base a resultados de error (por ejemplo, la velocidad de codificación y el nivel de distorsión). La unidad de procesamiento de predicción 41 puede proporcionar el bloque intra o intercodificado resultante al sumador 50 para generar datos de bloque residuales y al sumador 62 para reconstruir el bloque codificado para su uso como imagen de referencia.

[0098] La unidad de procesamiento de intrapredicción 46, dentro de la unidad de procesamiento de predicción 41, puede realizar la codificación con intrapredicción del bloque de vídeo actual con respecto a uno o más bloques vecinos en la misma trama o fragmento que el bloque actual que vaya a codificarse, para proporcionar compresión espacial. La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 dentro de la unidad de procesamiento de predicción 41 realizan la codificación interpredictiva del bloque de vídeo actual en relación con uno o más bloques predictivos en una o más imágenes de referencia, para proporcionar compresión temporal. La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 dentro de la unidad de procesamiento de predicción 41 pueden realizar la codificación predictiva entre capas del bloque de

vídeo actual en relación con uno o más bloques predictivos en una o más imágenes de referencia de una capa diferente.

5 **[0099]** La unidad de estimación de movimiento 42 se puede configurar para determinar el modo de
 interpredicción para un fragmento de vídeo de acuerdo con un patrón predeterminado para una secuencia de vídeo.
 El patrón predeterminado puede designar fragmentos de vídeo en la secuencia como fragmentos P, fragmentos B
 o fragmentos GPB. La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44
 pueden estar altamente integradas, pero se ilustran por separado para propósitos conceptuales. La estimación de
 10 movimiento, realizada por la unidad de estimación de movimiento 42, es el proceso de generación de vectores de
 movimiento (por ejemplo, vectores de movimiento temporales y vectores de movimiento de disparidad), que
 estiman el movimiento de los bloques de vídeo. Un vector de movimiento temporal, por ejemplo, puede indicar el
 desplazamiento de una PU de un bloque de vídeo dentro de una trama o imagen de vídeo actual en relación con
 un bloque predictivo dentro de una imagen de referencia. Un vector de movimiento de disparidad, por ejemplo,
 puede indicar el desplazamiento de una PU de un bloque de vídeo dentro de una trama o imagen de vídeo actual
 15 en relación con un bloque predictivo dentro de una imagen de referencia en una capa o vista diferente.

[0100] Un bloque predictivo es un bloque que se determina que coincide estrechamente con la PU del bloque de
 vídeo que se va a codificar en términos de diferencia de píxel, que se puede determinar mediante una suma de
 diferencia absoluta (SAD), una suma de diferencia al cuadrado (SSD) u otras métricas de diferencia. En algunos
 20 ejemplos, el codificador de vídeo 20 puede calcular valores para posiciones de píxel subentero de las imágenes
 de referencia almacenadas en la memoria intermedia de imágenes decodificadas 64. Por ejemplo, el codificador
 de vídeo 20 puede interpolar valores de posiciones de un cuarto de píxel, posiciones de un octavo de píxel u otras
 posiciones fraccionarias de píxel de la imagen de referencia. Por lo tanto, la unidad de estimación de movimiento
 42 puede realizar una búsqueda de movimiento en relación con las posiciones de píxel completo y las posiciones
 25 de píxel fraccionario, y emitir un vector de movimiento con una precisión de píxel fraccionario.

[0101] La unidad de estimación de movimiento 42 calcula un vector de movimiento para una PU de un bloque
 de vídeo en un fragmento intercodificado, comparando la posición de la PU con la posición de un bloque predictivo
 de una imagen de referencia. La imagen de referencia puede seleccionarse entre una primera lista de imágenes
 30 de referencia (lista 0) o una segunda lista de imágenes de referencia (lista 1), cada una de las cuales identifica una
 o más imágenes de referencia almacenadas en la memoria intermedia de imágenes decodificadas 64. La unidad
 de estimación de movimiento 42 envía el vector de movimiento calculado a la unidad de codificación por entropía
 56 y a la unidad de compensación de movimiento 44. Como parte de la determinación de un vector de movimiento
 para codificar un bloque de datos de vídeo, la unidad de estimación de movimiento 42 puede probar diversos
 35 modos de codificación que incluyen modos que implementan predicción de muestra entre capas, predicción de
 movimiento entre capas y/o una combinación de predicción de muestra entre capas y predicción de movimiento
 entre capas.

[0102] La compensación de movimiento, realizada por la unidad de compensación de movimiento 44, puede
 40 implicar extraer o generar el bloque predictivo en base al vector de movimiento determinado mediante estimación
 de movimiento, realizando posiblemente interpolaciones con una precisión de subpíxel. Tras recibir el vector de
 movimiento para la PU del bloque de vídeo actual, la unidad de compensación de movimiento 44 puede localizar
 el bloque predictivo al que apunta el vector de movimiento en una de las listas de imágenes de referencia. El
 45 codificador de vídeo 20 forma un bloque de vídeo residual restando los valores de píxel del bloque predictivo a los
 valores de píxel del bloque de vídeo actual que se está codificando, formando valores de diferencia de píxel. Los
 valores de diferencia de píxel forman datos residuales para el bloque, y pueden incluir componentes de diferencia
 tanto de luma como de croma. El sumador 50 representa el componente o los componentes que realizan esta
 operación de resta. La unidad de compensación de movimiento 44 también puede generar elementos sintácticos
 50 asociados a los bloques de vídeo y al fragmento de vídeo para su uso por el descodificador de vídeo 30 en la
 descodificación de los bloques de vídeo del fragmento de vídeo.

[0103] Algunas de las técnicas descritas en esta divulgación pueden implementarse en parte por la unidad de
 estimación de movimiento 42 y la unidad de compensación de movimiento 44. Por ejemplo, para una secuencia
 55 de vídeo codificada, la unidad de estimación de movimiento 42 puede determinar, en base a una compensación
 de distorsión de velocidad, por ejemplo, que la codificación de la secuencia de vídeo codificada no se beneficie del
 uso de un determinado tipo de predicción entre capas (que también incluiría señalar un valor para el elemento
 sintáctico de tipo de dependencia). En base a esa determinación, la unidad de estimación de movimiento 42 puede
 generar elementos sintácticos de acuerdo con las técnicas descritas anteriormente. Más en particular, en respuesta
 60 a la determinación de no usar un determinado tipo de predicción entre capas para la secuencia de vídeo codificada,
 la unidad de estimación de movimiento 42, posiblemente en combinación con otros componentes del codificador
 de vídeo 20, puede generar un valor para un primer elemento sintáctico que representa si se señala un elemento
 sintáctico de tipo de dependencia para una capa actual. En respuesta al valor para el primer elemento sintáctico
 que indica que el elemento sintáctico de tipo de dependencia no está señalado, la unidad de estimación de
 movimiento puede omitir la generación de valores para el elemento sintáctico de tipo de dependencia.
 65

[0104] Para reducir el número de bits necesarios para representar la información de movimiento de las PU, la unidad de estimación de movimiento 42 puede generar listas de candidatos para cada una de las PU de acuerdo con un modo de fusión o un proceso de predicción avanzada de vector de movimiento (AMVP). Cada candidato de una lista de candidatos para una PU puede indicar información de movimiento. La información de movimiento indicada por algunos de los candidatos de la lista de candidatos puede estar basada en la información de movimiento de otras PU. Por ejemplo, para el modo de fusión de HEVC, puede haber cinco localizaciones de candidatos espaciales y una localización de candidato temporal. La unidad de estimación de movimiento 42 también puede determinar uno o más candidatos de vectores de movimiento (por ejemplo, candidatos 3D HEVC o candidatos entre capas) en base a bloques correspondientes en diferentes vistas de capas. En algunos ejemplos, la unidad de estimación de movimiento 42 también puede generar candidatos adicionales combinando vectores de movimiento parcial de candidatos ya determinados, modificando candidatos o simplemente insertando vectores de movimiento cero como candidatos. Estos candidatos adicionales no se consideran candidatos originales y pueden denominarse candidatos virtuales. Para el modo de fusión, la unidad de estimación de movimiento 42 genera un índice para uno de los candidatos, y la información de movimiento para ese candidato se usa para codificar y descodificar el bloque. Para el modo de AMVP, la unidad de estimación de movimiento 42 genera un índice para uno de los candidatos, y la información de movimiento para ese candidato se usa como un predictor de movimiento. Por tanto, con el modo de AMVP, la unidad de estimación de movimiento 42 también genera información de la diferencia del vector de movimiento, y el predictor del vector de movimiento más la diferencia del vector de movimiento identifica el vector de movimiento usado para codificar el bloque.

[0105] Los tipos de candidatos que la unidad de compensación de movimiento 44 usa para construir la lista de candidatos pueden verse afectados por el tipo de dependencia entre una capa actual y una capa de referencia. Por ejemplo, si la predicción de muestra entre capas está habilitada, la unidad de estimación de movimiento 42 puede incluir, en una lista de candidatos, información de movimiento que apunta a un bloque en una imagen de referencia de una capa diferente (es decir, una capa de referencia). Como otro ejemplo, cuando la predicción de movimiento entre capas está habilitada, la unidad de estimación de movimiento 42 puede incluir la información de movimiento de un bloque en una capa diferente en una lista de candidatos para un bloque actual. La información de movimiento puede, por ejemplo, incluir un vector de movimiento (por ejemplo, un vector de movimiento de disparidad o un vector de movimiento temporal), y en algunos casos, también puede incluir información adicional, tal como un índice de imagen de referencia. Cuando se habilita una combinación de predicción de muestra entre capas y predicción de movimiento entre capas, un codificador de vídeo puede realizar tanto predicción de muestra entre capas como predicción de movimiento entre capas para determinar uno o más candidatos de una lista de candidatos.

[0106] Como se explica anteriormente, la manera en que la unidad de estimación de movimiento 42 construye una lista de candidatos para el modo de fusión o AMVP puede depender del tipo de dependencia entre capas que esté habilitado. Por ejemplo, si solo se habilita la predicción de muestra entre capas, entonces la unidad de estimación de movimiento 42 puede configurarse para no mirar bloques de diferentes capas para determinar candidatos. Si solo se habilita la predicción de movimiento entre capas, entonces la unidad de estimación de movimiento 42 puede configurarse para mirar bloques de diferentes capas para determinar candidatos, pero la unidad de estimación de movimiento 42 puede configurarse para incluir solo la información de movimiento de aquellos candidatos que usan vectores de movimiento temporal.

[0107] La unidad de procesamiento de intrapredicción 46 puede intrapredecir un bloque actual, como una alternativa a la interpredicción realizada por la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44, como se describe anteriormente. En particular, la unidad de procesamiento de intrapredicción 46 puede determinar un modo de intrapredicción a usar para codificar un bloque actual. En algunos ejemplos, la unidad de procesamiento de intrapredicción 46 puede codificar un bloque actual usando diversos modos de intrapredicción, por ejemplo, durante diferentes pases de codificación, y la unidad de procesamiento de intrapredicción 46 (o la unidad de selección de modo 40, en algunos ejemplos) puede seleccionar un modo adecuado de intrapredicción para su uso a partir de los modos probados. Por ejemplo, la unidad de procesamiento de intrapredicción 46 puede calcular valores de velocidad-distorsión usando un análisis de velocidad-distorsión para los diversos modos de intrapredicción sometidos a prueba, y seleccionar el modo de intrapredicción que tenga las mejores características de velocidad-distorsión entre los modos sometidos a prueba. El análisis de velocidad-distorsión determina, en general, una cantidad de distorsión (o error) entre un bloque codificado y un bloque original no codificado que se codificó para producir el bloque codificado, así como una velocidad de bits (es decir, un número de bits) usada para producir el bloque codificado. La unidad de procesamiento de intrapredicción 46 puede calcular relaciones a partir de las distorsiones y velocidades para los diversos bloques codificados, para determinar qué modo de intrapredicción presenta el mejor valor de velocidad-distorsión para el bloque.

[0108] En cualquier caso, tras seleccionar un modo de intrapredicción para un bloque, la unidad de procesamiento de intrapredicción 46 puede proporcionar información que indica el modo de intrapredicción seleccionado para el bloque a la unidad de codificación por entropía 56. La unidad de codificación por entropía 56 puede codificar la información que indica el modo de intrapredicción seleccionado de acuerdo con las técnicas de la presente divulgación. El codificador de vídeo 20 puede incluir, en el flujo de bits transmitido, datos de configuración, que pueden incluir una pluralidad de tablas de índices del modo de intrapredicción y una pluralidad

de tablas de índices del modo de intrapredicción modificadas (también denominadas tablas de correlación de palabras de código), definiciones de contextos de codificación para diversos bloques e indicaciones de un modo de intrapredicción más probable, una tabla de índices del modo de intrapredicción y una tabla de índices del modo de intrapredicción modificada para usar para cada uno de los contextos.

5

[0109] Después de que la unidad de procesamiento de predicción 41 genere el bloque predictivo para el bloque de vídeo actual, ya sea mediante interpredicción o intrapredicción, el codificador de vídeo 20 forma un bloque de vídeo residual restando el bloque predictivo al bloque de vídeo actual. Los datos de vídeo residuales del bloque residual se pueden incluir en una o más TU y aplicarse a la unidad de procesamiento de transformada 52. La unidad de procesamiento de transformada 52 transforma los datos de vídeo residuales en coeficientes de transformada residuales usando una transformada, tal como una transformada de coseno discreta (DCT) o una transformada conceptualmente similar. La unidad de procesamiento de transformada 52 puede convertir los datos de vídeo residuales de un dominio de píxel a un dominio de transformada, tal como un dominio de frecuencia.

10

15

[0110] La unidad de procesamiento de transformada 52 puede enviar los coeficientes de transformada resultantes a la unidad de cuantificación 54. La unidad de cuantificación 54 cuantifica los coeficientes de transformada para reducir más la velocidad de transferencia de bits. El proceso de cuantificación puede reducir la profundidad de bits asociada con algunos, o la totalidad, de los coeficientes. El grado de cuantificación se puede modificar ajustando un parámetro de cuantificación. En algunos ejemplos, la unidad de cuantificación 54 puede realizar a continuación una exploración de la matriz que incluye los coeficientes de transformada cuantificados. De forma alternativa, la unidad de codificación por entropía 56 puede realizar la exploración.

20

25

[0111] Tras la cuantificación, la unidad de codificación por entropía 56 codifica por entropía los coeficientes de transformada cuantificados. Por ejemplo, la unidad de codificación por entropía 56 puede realizar una codificación de longitud variable adaptativa al contexto (CAVLC), una codificación aritmética binaria adaptativa al contexto (CABAC), una codificación aritmética binaria adaptativa al contexto basada en la sintaxis (SBAC), una codificación por entropía por división de intervalos de probabilidad (PIPE) u otra metodología o técnica de codificación por entropía. Seguidamente a la codificación por entropía por la unidad de codificación por entropía 56, el flujo de bits codificado se puede transmitir al descodificador de vídeo 30, o archivarse para su posterior transmisión o recuperación por el descodificador de vídeo 30. La unidad de codificación por entropía 56 también puede codificar por entropía los vectores de movimiento y los otros elementos sintácticos para el fragmento de vídeo actual que se está codificando.

30

35

[0112] La unidad de cuantificación inversa 58 y la unidad de procesamiento de transformada inversa 60 aplican una cuantificación inversa y una transformada inversa, respectivamente, para reconstruir el bloque residual en el dominio de píxel para su posterior uso como bloque de referencia de una imagen de referencia. La unidad de compensación de movimiento 44 puede calcular un bloque de referencia añadiendo el bloque residual a un bloque predictivo de una de las imágenes de referencia dentro de una de las listas de imágenes de referencia. La unidad de compensación de movimiento 44 también puede aplicar uno o más filtros de interpolación al bloque residual reconstruido para calcular valores de píxeles fraccionarios, para su uso en la estimación del movimiento. El sumador 62 añade el bloque residual reconstruido al bloque de predicción compensado por movimiento, producido por la unidad de compensación de movimiento 44, para producir un bloque de referencia para su almacenamiento en la memoria intermedia de imágenes descodificadas 64. El bloque de referencia se puede usar por la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 como bloque de referencia para realizar una interpredicción de un bloque en una trama o imagen de vídeo subsiguiente.

40

45

50

[0113] De esta manera, el codificador de vídeo 20 de la FIG. 4 representa un ejemplo de un codificador de vídeo configurado para codificar datos de vídeo multicapa de acuerdo con las técnicas de esta divulgación. Para datos de vídeo codificados que comprenden múltiples capas, el codificador de vídeo 20 puede determinar si codifica las múltiples capas usando un solo tipo de dependencia o múltiples tipos de dependencia. Un tipo de dependencia identifica un tipo de dependencia de una capa actual en relación con una capa de referencia. Basado en la determinación de si codificar las múltiples capas usando un solo tipo de dependencia o múltiples tipos de dependencia, el codificador de vídeo 20 puede determinar un valor para un primer elemento sintáctico que representa si se señala un elemento sintáctico de tipo de dependencia para una capa actual. En respuesta a la determinación de codificar las múltiples capas usando el tipo de dependencia único, el codificador de vídeo 20 puede omitir la codificación del elemento sintáctico del tipo de dependencia. El primer elemento sintáctico puede incluirse en un conjunto de parámetros, y el codificador de vídeo 20 puede omitir la codificación del elemento sintáctico de tipo de dependencia al no incluir una instancia del elemento sintáctico de tipo de dependencia en el conjunto de parámetros. En respuesta a la determinación de codificar las múltiples capas usando los múltiples tipos de dependencia, el codificador de vídeo 20 puede codificar el elemento sintáctico de tipo de dependencia.

55

60

65

[0114] El mismo tipo de dependencia puede ser, por ejemplo, cualquiera de (1) solo predicción de muestra entre capas, (2) solo predicción de movimiento entre capas o (3) tanto predicción de muestra entre capas como predicción de movimiento entre capas. En este contexto, tener un mismo tipo de dependencia significa que todas las capas asociadas con el conjunto de parámetros tienen el mismo tipo de dependencia. Por tanto, si el codificador de vídeo 20 determina codificar las múltiples capas con el mismo tipo de dependencia, entonces todas las

combinaciones de capas tendrán un tipo de dependencia igual a, por ejemplo, el tipo (1) anterior, o todas las combinaciones de capas tendrán un tipo de dependencia igual al tipo (2) anterior, o todas las combinaciones de capas tendrán un tipo de dependencia igual al tipo (3) anterior. Si el codificador de vídeo 20 determina codificar las múltiples capas con múltiples tipos de dependencia, entonces algunas combinaciones de capas pueden tener un tipo de dependencia igual al tipo (1), mientras que otras combinaciones de capas pueden tener un tipo de dependencia igual al tipo (2), y aún otras combinaciones de capas pueden tener una dependencia igual al tipo (3).

[0115] La FIG. 5 es un diagrama de bloques que ilustra un descodificador de vídeo 30 de ejemplo que puede implementar las técnicas descritas en esta divulgación. El descodificador de vídeo 30 puede, por ejemplo, configurarse para descodificar datos de vídeo que están codificados usando la semántica descrita anteriormente. En el ejemplo de la FIG. 5, el descodificador de vídeo 30 incluye una memoria de datos de vídeo 79, una unidad de descodificación por entropía 80, una unidad de procesamiento de predicción 81, una unidad de cuantización inversa 86, una unidad de procesamiento de transformada inversa 88, un sumador 90, una unidad de filtro 91 y una memoria intermedia de imágenes codificadas 92. La unidad de procesamiento de predicción 81 incluye una unidad de compensación de movimiento 82 y una unidad de procesamiento de intrapredicción 84. En algunos ejemplos, el descodificador de vídeo 30 realiza un pase de descodificación, en general, recíproco al pase de codificación descrito con respecto al codificador de vídeo 20 de la FIG. 4.

[0116] La memoria de datos de vídeo 79 puede almacenar datos de vídeo, tales como un flujo de bits de vídeo codificado que los componentes del descodificador de vídeo 30 van a descodificar. Los datos de vídeo almacenados en la memoria de datos de vídeo 79 pueden obtenerse, por ejemplo, a partir del medio legible por ordenador 16, por ejemplo, desde una fuente de vídeo local, tal como una cámara, mediante comunicación de datos de vídeo por red cableada o inalámbrica, o accediendo a medios de almacenamiento físico de datos. La memoria de datos de vídeo 79 puede formar una memoria intermedia de imágenes codificadas (CPB) que almacena datos de vídeo codificados a partir de un flujo de bits de vídeo codificado. La memoria intermedia de imágenes descodificadas 162 puede ser una memoria de imágenes de referencia que almacena datos de vídeo de referencia para su uso en la descodificación de datos de vídeo mediante el descodificador de vídeo 30, por ejemplo, en los modos de intracodificación o intercodificación. La memoria de datos de vídeo 79 y la memoria intermedia de imágenes descodificadas 162 pueden estar formadas por cualquiera de una variedad de dispositivos de memoria, tales como memoria dinámica de acceso aleatorio (DRAM), incluyendo DRAM síncrona (SDRAM), RAM magnetorresistiva (MRAM), RAM resistiva (RRAM) u otros tipos de dispositivos de memoria. El mismo dispositivo de memoria u otros dispositivos de memoria separados pueden proporcionar una memoria de datos de vídeo 79 y una memoria intermedia de imágenes descodificadas 162. En diversos ejemplos, la memoria de datos de vídeo 79 puede estar en un chip con otros componentes del descodificador de vídeo 30, o fuera del chip con respecto a esos componentes.

[0117] Durante el procedimiento de descodificación, el descodificador de vídeo 30 recibe un flujo de bits de vídeo codificado, que representa bloques de vídeo de un fragmento de vídeo codificado y elementos sintácticos asociados, desde el codificador de vídeo 20. El descodificador de vídeo 30 puede recibir el flujo de bits de vídeo codificado desde una entidad de red 29. La entidad de red 29 puede ser, por ejemplo, un servidor, un MANE, un editor/empalmador de vídeo u otro dispositivo de este tipo configurado para implementar una o más de las técnicas descritas anteriormente. La entidad de red 29 puede incluir o no un codificador de vídeo, tal como el codificador de vídeo 20. Algunas de las técnicas descritas en la presente divulgación se pueden implementar por la entidad de red 29 antes de que la entidad de red 29 transmita el flujo de bits de vídeo codificado al descodificador de vídeo 30. En algunos sistemas de descodificación de vídeo, la entidad de red 29 y el descodificador de vídeo 30 pueden ser partes de dispositivos separados, mientras que en otros casos, la funcionalidad descrita con respecto a la entidad de red 29 se puede realizar por el mismo dispositivo que comprende el descodificador de vídeo 30.

[0118] La unidad de descodificación por entropía 80 del descodificador de vídeo 30 descodifica por entropía el flujo de bits para generar coeficientes cuantificados, vectores de movimiento y otros elementos sintácticos. La unidad de descodificación por entropía 80 reenvía los vectores de movimiento y otros elementos sintácticos a la unidad de procesamiento de predicción 81. El descodificador de vídeo 30 puede recibir los elementos sintácticos a nivel de fragmento de vídeo y/o a nivel de bloque de vídeo.

[0119] Cuando el fragmento de vídeo se codifica como un fragmento intracodificado (I), la unidad de procesamiento de intrapredicción 84 de la unidad de procesamiento de predicción 81 puede generar datos de predicción para un bloque de vídeo del fragmento de vídeo actual basándose en un modo de intrapredicción señalado, y datos de bloques previamente descodificados de la trama o imagen actual. Cuando la trama de vídeo se codifica como un fragmento intercodificado (es decir, B, P o GPB), la unidad de compensación de movimiento 82 de la unidad de procesamiento de predicción 81 genera bloques predictivos para un bloque de vídeo del fragmento de vídeo actual basándose en los vectores de movimiento y en otros elementos sintácticos recibidos desde la unidad de descodificación por entropía 80. Los bloques predictivos se pueden producir a partir de una de las imágenes de referencia dentro de una de las listas de imágenes de referencia. El descodificador de vídeo 30 puede construir las listas de tramas de referencia, la Lista 0 y la Lista 1, usando técnicas de construcción predeterminadas en base a imágenes de referencia almacenadas en la memoria intermedia de imágenes codificadas 92.

[0120] La unidad de compensación de movimiento 82 determina la información de predicción para un bloque de vídeo del fragmento de vídeo actual, analizando sintácticamente los vectores de movimiento y otros elementos sintácticos, y usa la información de predicción para producir los bloques predictivos para el bloque de vídeo actual que se está descodificando. Por ejemplo, la unidad de compensación de movimiento 82 usa algunos de los elementos sintácticos recibidos para determinar un modo de predicción (por ejemplo, intrapredicción o interpredicción) usado para codificar los bloques de vídeo del fragmento de vídeo, un tipo de fragmento de interpredicción (por ejemplo, un fragmento B, un fragmento P o un fragmento GPB), información de construcción para una o más de las listas de imágenes de referencia para el fragmento, vectores de movimiento para cada bloque de vídeo intercodificado del fragmento, el estado de interpredicción para cada bloque de vídeo intercodificado del fragmento y otra información para descodificar los bloques de vídeo en el fragmento de vídeo actual.

[0121] La unidad de compensación de movimiento 82 también puede realizar la interpolación basándose en filtros de interpolación. La unidad de compensación de movimiento 82 puede usar filtros de interpolación como los usados por el codificador de vídeo 20 durante la codificación de los bloques de vídeo para calcular valores interpolados para píxeles fraccionarios de los bloques de referencia. En este caso, la unidad de compensación de movimiento 82 puede determinar los filtros de interpolación usados por el codificador de vídeo 20 a partir de los elementos sintácticos recibidos y usar los filtros de interpolación para producir bloques predictivos.

[0122] La unidad de compensación de movimiento 82 puede descodificar los datos de vídeo recibidos usando múltiples intermodos, incluyendo la predicción entre capas. Por ejemplo, para un bloque de vídeo de una capa actual codificada que usa predicción de muestra entre capas, la unidad de compensación de movimiento 82 puede usar los elementos sintácticos recibidos para determinar un vector de movimiento de disparidad y usar el vector de movimiento de disparidad para identificar, en una capa de referencia, un bloque predictivo para el bloque de vídeo. Para un bloque de vídeo de una capa actual codificada que usa predicción de movimiento entre capas, la unidad de compensación de movimiento 82 puede usar los elementos sintácticos recibidos para localizar un bloque de referencia en una capa de referencia y, a partir del bloque de referencia, determinar la información de movimiento para el bloque de vídeo. Al usar la información de movimiento, la unidad de compensación de movimiento 82 puede identificar un bloque predictivo para el bloque de vídeo en una imagen de referencia de la capa actual.

[0123] Como parte de la intercodificación, la unidad de compensación de movimiento 82 puede implementar el modo de fusión y el modo de AMVP de la misma manera que la unidad de estimación de movimiento 42. Por tanto, la unidad de compensación de movimiento 82 puede configurarse para construir las mismas listas de candidatos que la unidad de estimación de movimiento 42, incluyendo la aplicación de cualquier restricción de candidatos basada en el tipo de dependencia entre capas habilitada.

[0124] La unidad de cuantificación inversa 86 cuantifica inversamente, es decir, descuantifica, los coeficientes de transformada cuantificados proporcionados en el flujo de bits y descodificados por la unidad de descodificación por entropía 80. El procedimiento de cuantificación inversa puede incluir el uso de un parámetro de cuantificación calculado por el codificador de vídeo 20 para cada bloque de vídeo del fragmento de vídeo para determinar un grado de cuantificación y, del mismo modo, un grado de cuantificación inversa que se debería aplicar. La unidad de procesamiento de transformada inversa 88 aplica una transformada inversa, por ejemplo, una DCT inversa, una transformada entera inversa o un procedimiento de transformada inversa conceptualmente similar, a los coeficientes de transformada para producir bloques residuales en el dominio de píxel.

[0125] Una vez que la unidad de compensación de movimiento 82 ha generado el bloque predictivo para el bloque de vídeo actual basándose en los vectores de movimiento y en otros elementos sintácticos, el descodificador de vídeo 30 forma un bloque de vídeo descodificado sumando los bloques residuales procedentes de la unidad de procesamiento de transformada inversa 88 a los correspondientes bloques predictivos generados por la unidad de compensación de movimiento 82. El sumador 90 representa el componente o los componentes que realizan esta operación de suma. Si se desea, también se pueden usar filtros de bucle (en el bucle de codificación o bien después del bucle de codificación) para suavizar las transiciones de píxeles o mejorar de otro modo la calidad de vídeo. La unidad de filtro 91 está destinada a representar uno o más filtros de bucle tales como un filtro de reducción del efecto bloque, un filtro de bucle adaptativo (ALF) y un filtro de desfase adaptativo de muestras (SAO). Aunque la unidad de filtro 91 que se muestra en la FIG. 5 es un filtro de bucle, en otras configuraciones la unidad de filtro 91 puede implementarse como un filtro postbucle. Los bloques de vídeo descodificados en una trama o imagen determinada se almacenan a continuación en la memoria intermedia de imágenes codificadas 92, que almacena imágenes de referencia usadas para una posterior compensación de movimiento. La memoria intermedia de imágenes codificadas 92 almacena también vídeo descodificado para su presentación posterior en un dispositivo de visualización, tal como el dispositivo de visualización 32 de la FIG. 1.

[0126] El descodificador de vídeo 30 de la FIG. 5 representa un ejemplo de descodificador de vídeo configurado para descodificar datos de vídeo multicapa de acuerdo con las técnicas de esta divulgación. Como parte de la descodificación de los datos de vídeo multicapa, el descodificador de vídeo 30 puede configurarse para recibir un valor para un primer elemento sintáctico que representa si se señala un elemento sintáctico de tipo de

dependencia para una capa actual. El elemento sintáctico de tipo de dependencia identifica un tipo de dependencia de una capa actual en relación con una o más capas de referencia. En respuesta al valor para el primer elemento sintáctico que indica que el elemento sintáctico de tipo de dependencia no está señalado, el descodificador de vídeo 30 puede determinar que el tipo de dependencia de la capa actual en relación con la capa de referencia es un tipo predeterminado y descodificar un bloque de la capa actual usando predicción entre capas conforme al tipo predeterminado.

[0127] Para determinar que el tipo de dependencia de la capa actual en relación con la capa de referencia es el tipo predeterminado, el descodificador de vídeo 30 puede configurarse para inferir un valor del elemento sintáctico de tipo de dependencia. Para determinar que el tipo de dependencia de la capa actual en relación con la capa de referencia es el tipo predeterminado, el descodificador de vídeo 30 puede determinar el tipo de dependencia entre capas sin descodificar una instancia del elemento sintáctico del tipo de dependencia para la capa actual. El tipo predeterminado puede ser, por ejemplo, uno de (1) predicción de muestra entre capas, (2) predicción de movimiento entre capas o (3) una combinación de predicción de muestra entre capas y predicción de movimiento entre capas. En casos donde el tipo predeterminado es la predicción de muestra entre capas, el descodificador de vídeo 30 puede descodificar el bloque de la capa actual usando la predicción entre capas identificando, en la capa de referencia, un bloque predictivo para el bloque de la capa actual.

[0128] El descodificador de vídeo 30 puede recibir el primer elemento sintáctico como parte de un conjunto de parámetros, y en respuesta al valor para el primer elemento sintáctico que indica que el elemento sintáctico del tipo de dependencia no está señalado, el descodificador de vídeo 30 puede determinar que el tipo de dependencia de la capa actual en relación con todas las otras capas asociadas con el conjunto de parámetros es el tipo predeterminado. Adicionalmente, en respuesta al valor para el primer elemento sintáctico que indica que el elemento sintáctico de tipo de dependencia está señalado, el descodificador de vídeo 30 puede recibir un valor para el elemento sintáctico de tipo de dependencia, determinar el tipo de dependencia de la capa actual en relación con la capa de referencia en base al valor para el elemento sintáctico de tipo de dependencia y descodificar el bloque de la capa actual usando predicción entre capas conforme al tipo determinado. El valor del elemento sintáctico de tipo de dependencia puede indicar que el tipo de dependencia de la capa actual en relación con la capa de referencia es uno de (1) predicción de muestra entre capas, (2) predicción de movimiento entre capas o (3) una combinación de predicción de muestra entre capas y predicción de movimiento entre capas.

[0129] La FIG. 6 es un diagrama de bloques que ilustra un conjunto de ejemplo de dispositivos que forman parte de la red 100. En este ejemplo, la red 100 incluye los dispositivos de encaminamiento 104A, 104B (dispositivos de encaminamiento 104) y el dispositivo de transcodificación 106. Los dispositivos de encaminamiento 104 y el dispositivo de transcodificación 106 están concebidos para representar un pequeño número de dispositivos que pueden formar parte de la red 100. Otros dispositivos de red, tales como conmutadores, concentradores, pasarelas, cortafuegos, puentes y otros dispositivos de ese tipo también pueden estar incluidos dentro de la red 100. Además, pueden proporcionarse dispositivos de red adicionales a lo largo de un trayecto de red entre un dispositivo servidor 102 y un dispositivo cliente 108. El dispositivo servidor 102 puede corresponder a un dispositivo de origen 12 (FIG. 1), mientras que el dispositivo cliente 108 puede corresponder a un dispositivo de destino 14 (FIG. 1), en algunos ejemplos.

[0130] En general, los dispositivos de encaminamiento 104 implementan uno o más protocolos de encaminamiento para intercambiar datos de red a través de la red 100. En algunos ejemplos, los dispositivos de encaminamiento 104 pueden estar configurados para realizar operaciones de apoderado (*proxy*) o de memoria caché. Por lo tanto, en algunos ejemplos, los dispositivos de encaminamiento 104 pueden denominarse dispositivos de apoderado. En general, los dispositivos de encaminamiento 104 ejecutan protocolos de encaminamiento para descubrir rutas a través de la red 100. Al ejecutar dichos protocolos de encaminamiento, un dispositivo de encaminamiento 104B puede descubrir una ruta de red desde sí mismo hasta el dispositivo servidor 102 por medio del dispositivo de encaminamiento 104A.

[0131] Dispositivos de red, tales como dichos dispositivos de encaminamiento 104 y dispositivo de transcodificación 106, pueden implementar las técnicas de esta divulgación, aunque un dispositivo cliente 108 también las puede implementar. De esta manera, los dispositivos de encaminamiento 104, el dispositivo de transcodificación 106 y el dispositivo cliente 108 representan ejemplos de dispositivos configurados para realizar las técnicas de esta divulgación. Por otro lado, los dispositivos de la FIG. 1 y el codificador 20 ilustrado en la FIG. 4 y el descodificador 30 ilustrado en la FIG. 5 son también dispositivos a modo de ejemplo que pueden estar configurados para realizar las técnicas de esta divulgación.

[0132] Por ejemplo, el dispositivo servidor 102 puede incluir un codificador de vídeo para codificar una imagen o unas imágenes que se encuentran después de un punto de acceso aleatorio o un punto de adaptación de flujo, tal como un punto de conmutación de capa temporal, u otro punto de adaptación de flujo. Por ejemplo, este punto podría ser un punto de conmutación para la adaptación de la velocidad de bits, la velocidad de tramas (es decir, el punto de conmutación de capa temporal) o la resolución espacial. Asimismo, el dispositivo cliente 108 puede descodificar una imagen o imágenes que están después de un punto de acceso aleatorio o punto de adaptación de flujo, tal como un punto de conmutación de capa temporal.

[0133] La FIG. 7 es un diagrama de flujo que ilustra un procedimiento ejemplar de codificación de datos de vídeo multicapa de acuerdo con las técnicas de esta divulgación. Las técnicas de la FIG. 7 se describirán con respecto al codificador de vídeo 20, aunque las técnicas de la FIG. 7 no se limitan a ningún tipo particular de codificador de vídeo. En el ejemplo de la FIG. 7, el codificador de vídeo 20 está configurado para codificar datos de vídeo multicapa. Para datos de vídeo codificados que incluyen múltiples capas, el codificador de vídeo 20 determina si se codifican las múltiples capas usando un mismo tipo de dependencia para todas las capas de las múltiples capas o múltiples tipos de dependencia (170). Un tipo de dependencia identifica un tipo de dependencia de una capa actual en relación con una capa de referencia. Basado en la determinación de si codificar las múltiples capas usando el mismo tipo de dependencia o múltiples tipos de dependencia, el codificador de vídeo 20 determina un valor para un primer elemento sintáctico que representa si se señala un elemento sintáctico de tipo de dependencia para una capa actual (172). El primer elemento sintáctico puede incluirse, por ejemplo, en un conjunto de parámetros, y el codificador de vídeo 20 puede omitir la codificación del elemento sintáctico de tipo de dependencia al no incluir una instancia del elemento sintáctico de tipo de dependencia en el conjunto de parámetros. En respuesta a la determinación de codificar las múltiples capas usando el mismo tipo de dependencia (174, sí), el codificador de vídeo 20 omite la codificación del elemento sintáctico de tipo de dependencia (176). En respuesta a la determinación de codificar las múltiples capas usando los múltiples tipos de dependencia (174, no), el codificador de vídeo 20 codifica el elemento sintáctico de tipo de dependencia (178). El mismo tipo de dependencia puede ser, por ejemplo, uno de (1) predicción de muestra entre capas, (2) predicción de movimiento entre capas o (3) una combinación de predicción de muestra entre capas y predicción de movimiento entre capas.

[0134] La FIG. 8 es un diagrama de flujo que ilustra un procedimiento ejemplar de descodificación de datos de vídeo multicapa de acuerdo con las técnicas de esta divulgación. Las técnicas de la FIG. 8 se describirán con respecto al descodificador de vídeo 30, aunque las técnicas de la FIG. 8 no se limitan a ningún tipo particular de descodificador de vídeo. De acuerdo con las técnicas de la FIG. 7, el descodificador de vídeo 30 puede descodificar datos de vídeo multicapa. El descodificador de vídeo 30 puede recibir un valor para un primer elemento sintáctico que representa si un elemento sintáctico de tipo de dependencia para una capa actual está señalado (180). El elemento sintáctico de tipo de dependencia identifica un tipo de dependencia de una capa actual en relación con una capa de referencia. En respuesta al valor para el primer elemento sintáctico que indica que el elemento sintáctico de tipo de dependencia no está señalado (182, sí), el descodificador de vídeo 30 puede determinar que el tipo de dependencia de la capa actual en relación con la capa de referencia es un tipo predeterminado (184) y descodificar un bloque de la capa actual usando predicción entre capas conforme al tipo predeterminado (186). En respuesta al valor para el primer elemento sintáctico que indica que el elemento sintáctico de tipo de dependencia está señalado (182, no), el descodificador de vídeo 30 puede recibir un valor para el elemento sintáctico de tipo de dependencia (183), determinar el tipo de dependencia de la capa actual en relación con la capa de referencia en base al valor para el elemento sintáctico de tipo de dependencia (185) y descodificar el bloque de la capa actual usando predicción entre capas conforme al tipo determinado (187). El valor del elemento sintáctico de tipo de dependencia puede indicar que el tipo de dependencia de la capa actual en relación con la capa de referencia es uno de (1) predicción de muestra entre capas, (2) predicción de movimiento entre capas o (3) una combinación de predicción de muestra entre capas y predicción de movimiento entre capas.

[0135] El descodificador de vídeo 30 puede, por ejemplo, determinar que el tipo de dependencia de la capa actual en relación con la capa de referencia es el tipo predeterminado al inferir un valor del elemento sintáctico del tipo de dependencia. De forma alternativa o adicionalmente, el descodificador de vídeo 30 puede determinar que el tipo de dependencia de la capa actual en relación con la capa de referencia es el tipo predeterminado determinando el tipo de dependencia entre capas sin descodificar una instancia del elemento sintáctico del tipo de dependencia para la capa actual. El tipo predeterminado puede ser, por ejemplo, uno de (1) predicción de muestra entre capas, (2) predicción de movimiento entre capas o (3) una combinación de predicción de muestra entre capas y predicción de movimiento entre capas. En algunos ejemplos, el tipo predeterminado puede ser predicción de muestra entre capas, y el descodificador de vídeo 30 puede descodificar el bloque de la capa actual usando predicción entre capas identificando, en la capa de referencia, un bloque predictivo para el bloque de la capa actual.

[0136] El descodificador de vídeo 30 puede recibir el primer elemento sintáctico, por ejemplo, como parte de un conjunto de parámetros, y en respuesta al valor para el primer elemento sintáctico que indica que el elemento sintáctico del tipo de dependencia no está señalado, el descodificador de vídeo 30 puede determinar que el tipo de dependencia de la capa actual en relación con todas las otras capas asociadas con el conjunto de parámetros es el tipo predeterminado.

[0137] En uno o más ejemplos, las funciones descritas se pueden implementar en hardware, software, firmware o cualquier combinación de los mismos. Si se implementan en software, las funciones se pueden almacenar en o transmitir sobre un medio legible por ordenador como una o más instrucciones o código, y ejecutarse por una unidad de procesamiento basada en hardware. Los medios legibles por ordenador pueden incluir medios de almacenamiento legibles por ordenador, que corresponden a un medio tangible tal como unos medios de almacenamiento de datos, o unos medios de comunicación que incluyen cualquier medio que facilita la transferencia de un programa informático de un lugar a otro, por ejemplo, de acuerdo con un protocolo de comunicación. De esta manera, los medios legibles por ordenador pueden corresponder, en general, a (1) medios

de almacenamiento tangibles legibles por ordenador que son no transitorios o (2) un medio de comunicación tal como una señal o una onda portadora. Los medios de almacenamiento de datos pueden ser cualquier medio disponible al que se pueda acceder desde uno o más ordenadores o uno o más procesadores para recuperar instrucciones, código y/o estructuras de datos para la implementación de las técnicas descritas en la presente divulgación. Un producto de programa informático puede incluir un medio legible por ordenador.

[0138] A modo de ejemplo, y no de limitación, dichos medios de almacenamiento legibles por ordenador pueden comprender RAM, ROM, EEPROM, CD-ROM u otro almacenamiento en disco óptico, almacenamiento en disco magnético u otros dispositivos de almacenamiento magnético, memoria flash o cualquier otro medio que se pueda usar para almacenar código de programa deseado en forma de instrucciones o estructuras de datos y al que se pueda acceder mediante un ordenador. Además, cualquier conexión recibe apropiadamente la denominación de medio legible por ordenador. Por ejemplo, si las instrucciones se transmiten desde un sitio web, un servidor u otra fuente remota usando un cable coaxial, un cable de fibra óptica, un par trenzado, una línea de abonado digital (DSL) o unas tecnologías inalámbricas tales como infrarrojos, radio y microondas, entonces el cable coaxial, el cable de fibra óptica, el par trenzado, la DSL o las tecnologías inalámbricas tales como infrarrojos, radio y microondas están incluidos en la definición de medio. Sin embargo, se debe entender que los medios de almacenamiento legibles por ordenador y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales ni otros medios transitorios, sino que, en cambio, se refieren a medios de almacenamiento tangibles no transitorios. El término disco, como se usa en el presente documento, incluye el disco compacto (CD), el disco láser, el disco óptico, el disco versátil digital (DVD), el disco flexible y el disco Blu-ray, donde algunos discos reproducen normalmente los datos magnéticamente, mientras que otros discos reproducen los datos ópticamente con láseres. Las combinaciones de los anteriores también se deben incluir dentro del alcance de los medios legibles por ordenador.

[0139] Uno o más procesadores, tales como uno o más procesadores de señales digitales (DSP), microprocesadores de propósito general, circuitos integrados específicos de la aplicación (ASIC), matrices lógicas programables in situ (FPGA) u otros circuitos lógicos integrados o discretos equivalentes pueden ejecutar las instrucciones. Por consiguiente, el término "procesador", como se usa en el presente documento, se puede referir a cualquiera de las estructuras anteriores o a cualquier otra estructura adecuada para la implementación de las técnicas descritas en el presente documento. Además, en algunos aspectos, la funcionalidad descrita en el presente documento se puede proporcionar dentro de módulos de hardware y/o de software dedicados configurados para codificar y decodificar, o incorporar en un códec combinado. Asimismo, las técnicas se podrían implementar por completo en uno o más circuitos o elementos lógicos.

[0140] Las técnicas de esta divulgación se pueden implementar en una amplia variedad de dispositivos o aparatos, incluyendo un teléfono inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). En esta divulgación se describen diversos componentes, módulos o unidades para destacar aspectos funcionales de dispositivos configurados para realizar las técnicas divulgadas, pero no se requiere necesariamente su realización mediante diferentes unidades de hardware. En su lugar, como se ha descrito anteriormente, diversas unidades se pueden combinar en una unidad de hardware de códec o proporcionar mediante un grupo de unidades de hardware interoperativas, que incluye uno o más procesadores como se describe anteriormente, junto con software y/o firmware adecuados.

[0141] Se han descrito diversos ejemplos. Estos y otros ejemplos están dentro del alcance de las siguientes reivindicaciones.

REIVINDICACIONES

1. Un procedimiento de descodificación de datos de vídeo de múltiples capas, comprendiendo el procedimiento:

5 recibir (180) un valor para un primer elemento sintáctico que representa si se señala un elemento sintáctico de tipo de dependencia, en el que, cuando se señala el elemento sintáctico de tipo de dependencia, el elemento sintáctico de tipo de dependencia identifica un tipo de dependencia habilitada para cada capa de los datos de vídeo multicapa en relación con cada capa de referencia, siendo el tipo de dependencia identificado por el elemento sintáctico de tipo de dependencia uno de (1) predicción de muestra entre capas, 10 (2) predicción de movimiento entre capas o (3) una combinación de predicción de muestra entre capas y predicción de movimiento entre capas;

15 si el valor para el primer elemento sintáctico indica que el elemento sintáctico de tipo de dependencia está señalado, recibir (183) valores para el elemento sintáctico de tipo de dependencia, determinar (185) el tipo de dependencia habilitada para una capa actual en relación con una capa de referencia basada en un valor respectivo del elemento sintáctico de tipo de dependencia, y descodificar (187) un bloque de la capa actual usando predicción entre capas conforme al tipo de dependencia; o

20 si el valor para el primer elemento sintáctico indica que el elemento sintáctico de tipo de dependencia no está señalado, recibir una indicación señalizada por separado de un tipo de dependencia único que es el mismo para todas las combinaciones de capas actuales y capas de referencia, en el que la indicación señalizada por separado indica cuál de (1) predicción de muestra entre capas, (2) predicción de movimiento entre capas o (3) una combinación de predicción de muestra entre capas y predicción de movimiento entre 25 capas, es el único tipo de dependencia y determinar (184) que un tipo de dependencia de cualquier capa actual en relación con una capa de referencia es el único tipo de dependencia, y descodificar (186) un bloque de la capa actual usando predicción entre capas conforme al único tipo de dependencia.

30 2. El procedimiento de la reivindicación 1, en el que el tipo de dependencia habilitado para todas las capas en relación con cada capa de referencia comprende uno de un primer tipo de dependencia donde la capa de referencia se usa para la predicción de muestra entre capas de la capa actual pero no para la predicción de movimiento entre capas de la capa actual, un segundo tipo de dependencia donde la capa de referencia se usa para la predicción de movimiento entre capas de la capa actual pero no para la predicción de muestra entre capas de la capa actual, o un tercer tipo de dependencia donde la capa de referencia se usa tanto para la predicción de muestras entre 35 capas como para la predicción de movimiento entre capas de la capa actual.

40 3. El procedimiento de la reivindicación 1, en el que el tipo de dependencia habilitado para todas las capas en relación con cada capa de referencia comprende la predicción de muestra entre capas, y en el que descodificar el bloque de la capa actual usando la predicción entre capas comprende identificar, en la capa de referencia, un bloque predictivo para el bloque de la capa actual.

4. El procedimiento de la reivindicación 1, que comprende además:

45 recibir el primer elemento sintáctico como parte de un conjunto de parámetros, que es preferentemente un conjunto de parámetros de vídeo;

en respuesta al valor del primer elemento sintáctico que indica que el elemento sintáctico de tipo de dependencia no está señalado, determinando que el mismo tipo de dependencia está habilitado para todas las capas en relación con todas las otras capas asociadas con el conjunto de parámetros.

50 5. Un procedimiento de codificación de datos de vídeo multicapa, comprendiendo el procedimiento:

55 para datos de vídeo codificados que comprenden múltiples capas, determinar si se codifican las múltiples capas usando un solo tipo de dependencia para todas las capas de las múltiples capas o usando múltiples tipos de dependencia para las múltiples capas, en el que un tipo de dependencia identifica un tipo de dependencia habilitada para una capa actual en relación con una capa de referencia, y es uno de (1) predicción de muestra entre capas, (2) predicción de movimiento entre capas o (3) una combinación de predicción de muestra entre capas y predicción de movimiento entre capas;

60 basado en la determinación de si se codifican las múltiples capas usando el único tipo de dependencia o múltiples tipos de dependencia, determinar un valor para un primer elemento sintáctico que represente si se señala un elemento sintáctico de tipo de dependencia, en el que el elemento sintáctico de tipo de dependencia identifica un tipo de dependencia habilitada para cada capa de los datos de vídeo multicapa en relación con cada capa de referencia;

65 codificar el elemento sintáctico de tipo de dependencia si se determina codificar las múltiples capas usando múltiples tipos de dependencia; y

- 5 si se determina codificar las múltiples capas usando el único tipo de dependencia, omitir la codificación del elemento sintáctico de tipo de dependencia y señalar por separado una indicación de un único tipo de dependencia que es el mismo para todas las combinaciones de capas actuales y capas de referencia, en el que la indicación señalizada por separado indica cuál de (1) predicción de muestra entre capas, (2) predicción de movimiento entre capas o (3) una combinación de predicción de muestra entre capas y predicción de movimiento entre capas, es el único tipo de dependencia.
- 10 **6.** El procedimiento de la reivindicación 5, en el que el primer elemento sintáctico se incluye en un conjunto de parámetros y en el que omitir la codificación del elemento sintáctico de tipo de dependencia comprende no incluir una instancia del elemento sintáctico de tipo de dependencia en el conjunto de parámetros.
- 15 **7.** El procedimiento de la reivindicación 5, en el que el mismo tipo de dependencia comprende uno de un primer tipo de dependencia donde la capa de referencia se usa para la predicción de muestra entre capas de la capa actual pero no para la predicción de movimiento entre capas de la capa actual, un segundo tipo de dependencia donde la capa de referencia se usa para la predicción de movimiento entre capas de la capa actual pero no para la predicción de muestra entre capas de la capa actual, o un tercer tipo de dependencia donde la capa de referencia se usa tanto para la predicción de muestras entre capas como para la predicción de movimiento entre capas de la capa actual.
- 20 **8.** Un aparato que realiza descodificación de vídeo, comprendiendo el aparato:
- una memoria que almacena datos de vídeo; y
- 25 un codificador de vídeo que comprende uno o más procesadores, configurados para:
- 30 recibir un valor para un primer elemento sintáctico que representa si se señala un elemento sintáctico de tipo de dependencia, en el que, cuando se señala la sintaxis de tipo de dependencia, el elemento sintáctico de tipo de dependencia identifica un tipo de dependencia habilitada para cada capa de los datos de vídeo multicapa en relación con cada capa de referencia, siendo el tipo de dependencia identificado por el elemento sintáctico de tipo de dependencia uno de (1) predicción de muestra entre capas, (2) predicción de movimiento entre capas o (3) una combinación de predicción de muestra entre capas y predicción de movimiento entre capas;
- 35 si el valor para el primer elemento sintáctico indica que el elemento sintáctico de tipo de dependencia está señalado, recibir valores para el elemento sintáctico de tipo de dependencia, determinar el tipo de dependencia habilitada para cualquier capa actual en relación con una capa de referencia basada en un valor respectivo del elemento sintáctico de tipo de dependencia, y descodificar un bloque de la capa actual usando predicción entre capas conforme al tipo de dependencia; o
- 40 si el valor para el primer elemento sintáctico indica que el elemento sintáctico de tipo de dependencia no está señalado, recibir una indicación señalizada por separado de un tipo de dependencia único que es el mismo para todas las combinaciones de capas actuales y capas de referencia, en el que la indicación señalizada por separado indica cuál de (1) predicción de muestra entre capas, (2) predicción de movimiento entre capas o (3) una combinación de predicción de muestra entre capas y predicción de movimiento entre capas, es el único tipo de dependencia y determinar que un tipo de dependencia de una capa actual en relación con una capa de referencia es el único tipo de dependencia, y descodificar un bloque de la capa actual usando predicción entre capas conforme al tipo de dependencia determinado.
- 50 **9.** El aparato de la reivindicación 8, en el que, cuando el valor para el elemento sintáctico indica que el tipo de dependencia no está señalado, el valor para el primer elemento sintáctico identifica además el tipo de dependencia habilitado para todas las capas en relación con cada capa de referencia.
- 55 **10.** El aparato de la reivindicación 8, en el que el tipo de dependencia habilitado para todas las capas en relación con cada capa de referencia comprende uno de un primer tipo de dependencia donde la capa de referencia se usa para la predicción de muestra entre capas de la capa actual pero no para la predicción de movimiento entre capas de la capa actual, un segundo tipo de dependencia donde la capa de referencia se usa para la predicción de movimiento entre capas de la capa actual pero no para la predicción de muestra entre capas de la capa actual, o un tercer tipo de dependencia donde la capa de referencia se usa tanto para la predicción de muestras entre capas como para la predicción de movimiento entre capas de la capa actual.
- 60 **11.** El aparato de la reivindicación 8, en el que el tipo de dependencia habilitado para todas las capas en relación con cada capa de referencia comprende la predicción de muestra entre capas, y en el que descodificar el bloque de la capa actual usando la predicción entre capas comprende identificar, en la capa de referencia, un bloque predictivo para el bloque de la capa actual.
- 65

12. El aparato de la reivindicación 8, en el que el uno o más procesadores están configurados, además, para:

5 recibir el primer elemento sintáctico como parte de un conjunto de parámetros, que es preferentemente un conjunto de parámetros de vídeo;

10 en respuesta al valor del primer elemento sintáctico que indica que el elemento sintáctico de tipo de dependencia no está señalado, determinar que el mismo tipo de dependencia está habilitado para todas las capas en relación con todas las otras capas asociadas con el conjunto de parámetros.

13. Un aparato que realiza codificación de vídeo, comprendiendo el aparato:

15 una memoria que almacena datos de vídeo; y

un codificador de vídeo que comprende uno o más procesadores, configurados para:

20 para datos de vídeo codificados que comprenden múltiples capas, determinar si se codifican las múltiples capas usando un solo tipo de dependencia para todas las capas de las múltiples capas o usando múltiples tipos de dependencia para las múltiples capas, en el que un tipo de dependencia identifica un tipo de dependencia habilitada para una capa actual en relación con una capa de referencia, y es uno de (1) predicción de muestra entre capas, (2) predicción de movimiento entre capas o (3) una combinación de predicción de muestra entre capas y predicción de movimiento entre capas;

25 basado en la determinación de si se codifican las múltiples capas usando el único tipo de dependencia o múltiples tipos de dependencia, determinar un valor para un primer elemento sintáctico que represente si se señala un elemento sintáctico de tipo de dependencia, en el que el elemento sintáctico de tipo de dependencia identifica un tipo de dependencia habilitada para cada capa de los datos de vídeo multicapa en relación con cada capa de referencia;

30 codificar el elemento sintáctico de tipo de dependencia si se determina codificar las múltiples capas usando múltiples tipos de dependencia; y

35 si se determina codificar las múltiples capas usando el único tipo de dependencia, omitir la codificación del elemento sintáctico de tipo de dependencia y señalar por separado una indicación de un único tipo de dependencia que es el mismo para todas las combinaciones de capas actuales y capas de referencia, en el que la indicación señalizada por separado indica cuál de (1) predicción de muestra entre capas, (2) predicción de movimiento entre capas o (3) una combinación de predicción de muestra entre capas y predicción de movimiento entre capas, es el único tipo de dependencia.

14. El aparato de la reivindicación 8 o la reivindicación 13, en el que el aparato comprende al menos uno de:

40 un circuito integrado;

45 un microprocesador; o

un dispositivo de comunicación inalámbrica que incluye un codificador de vídeo.

15. Un medio de almacenamiento legible por ordenador que almacena instrucciones que, cuando son ejecutadas por uno o más procesadores, hacen que el uno o más procesadores realicen el procedimiento de cualquiera de las reivindicaciones 1 a 7.

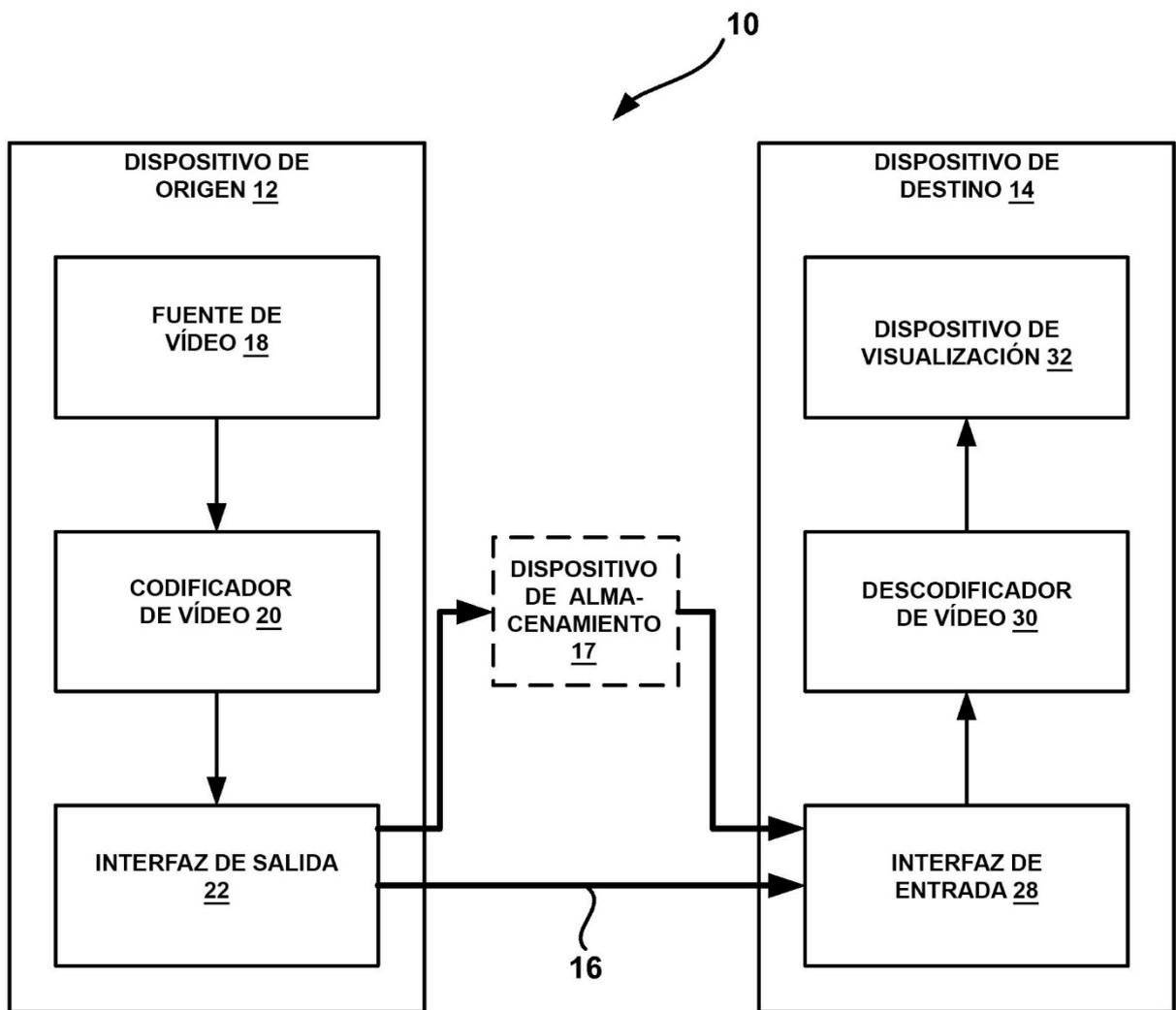


FIG. 1

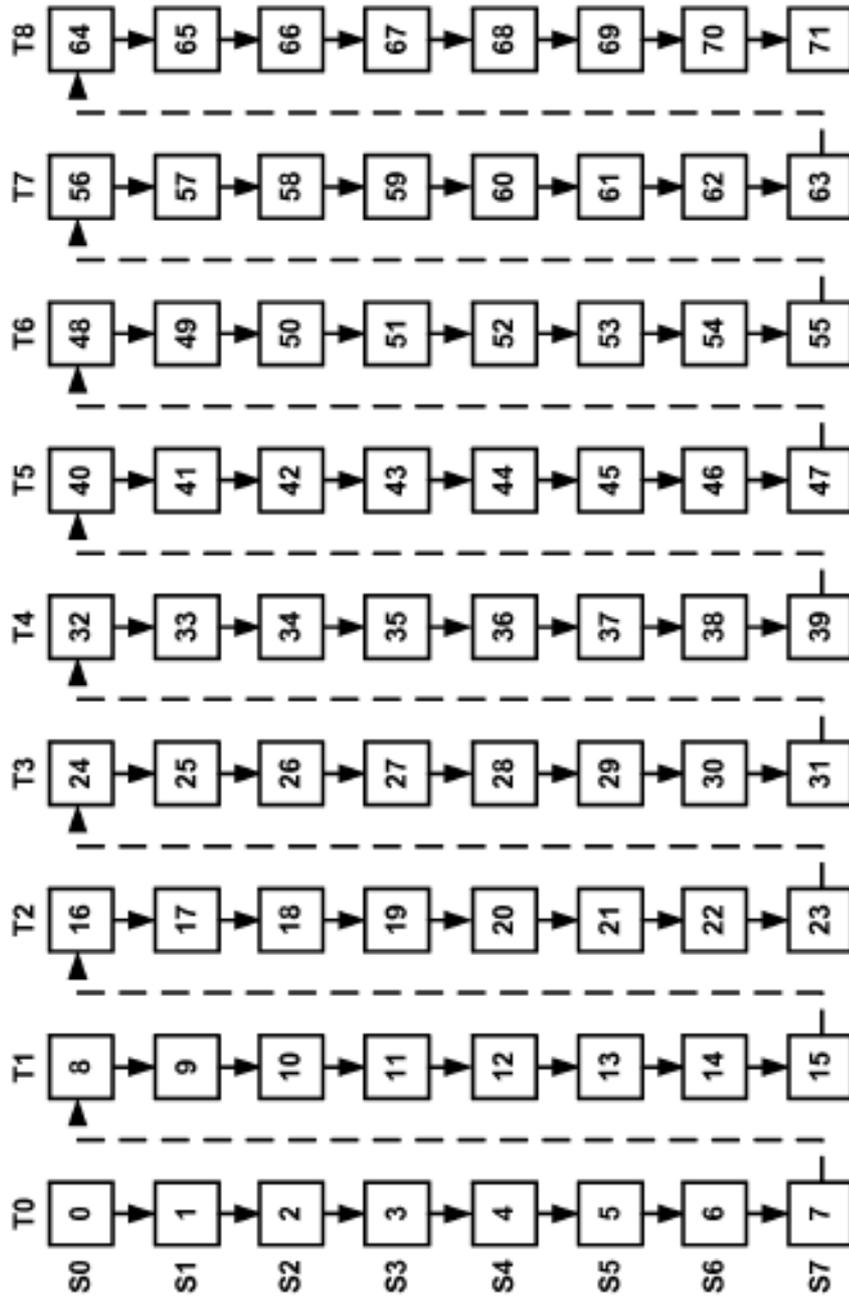


FIG. 2

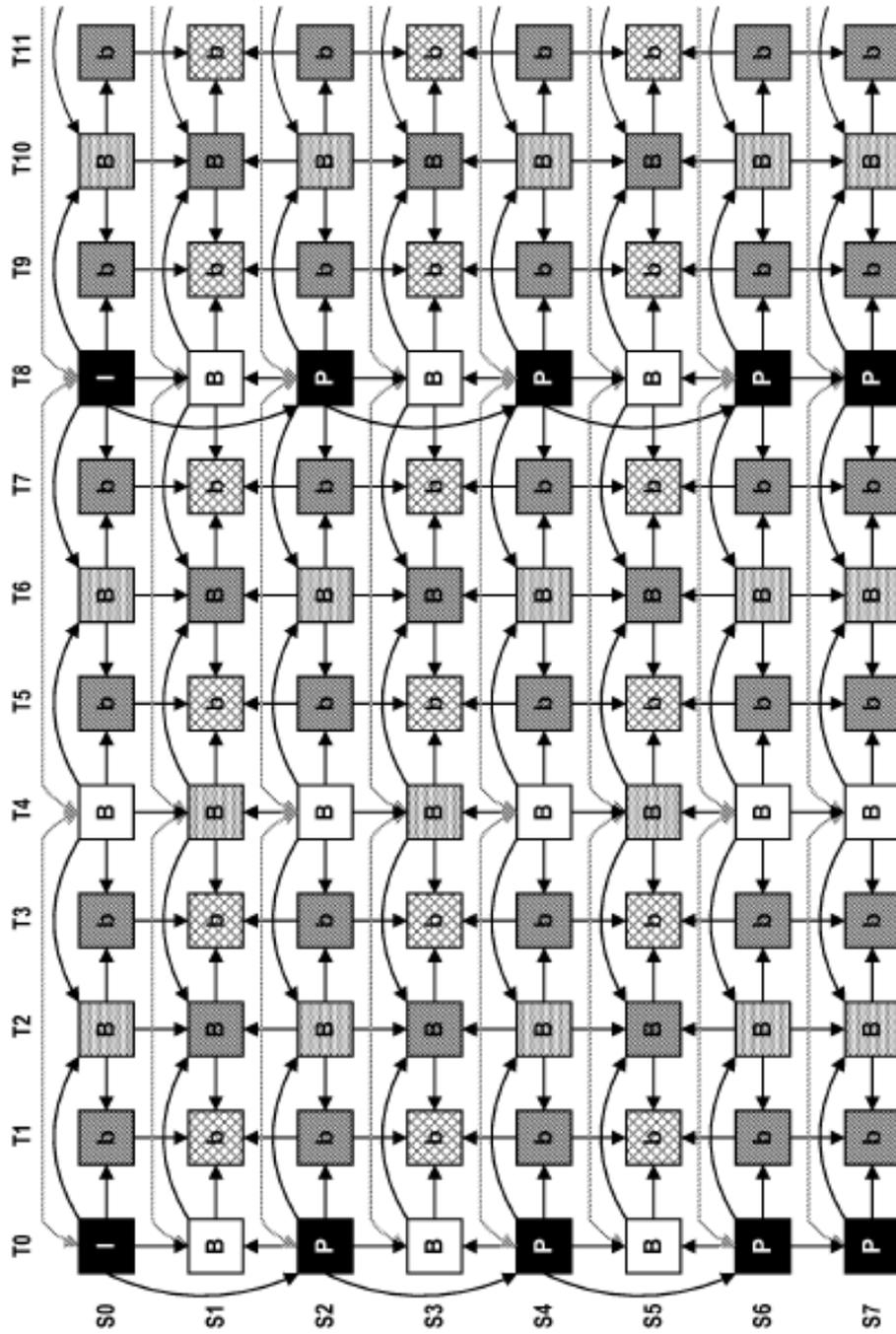


FIG. 3

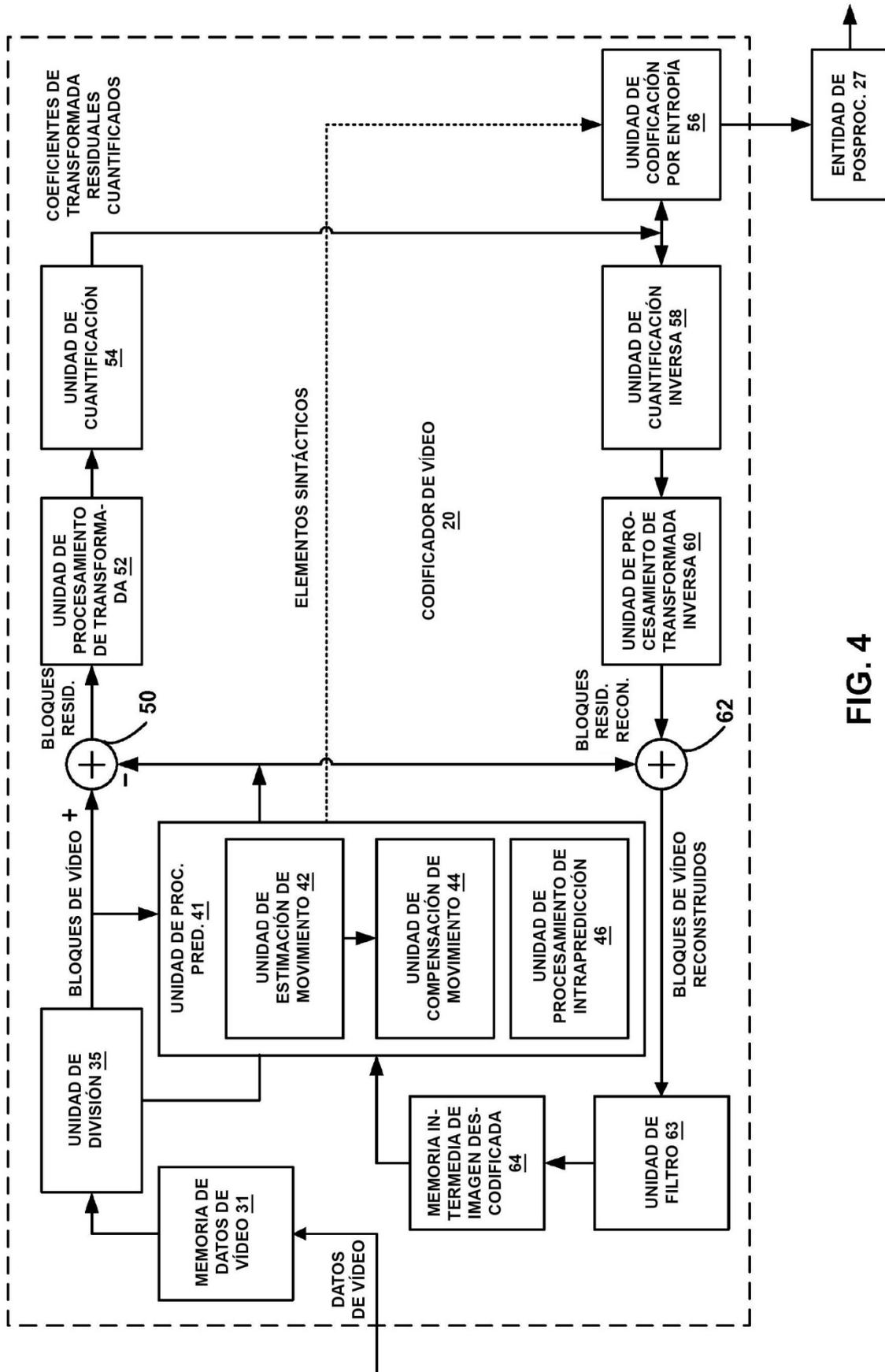


FIG. 4

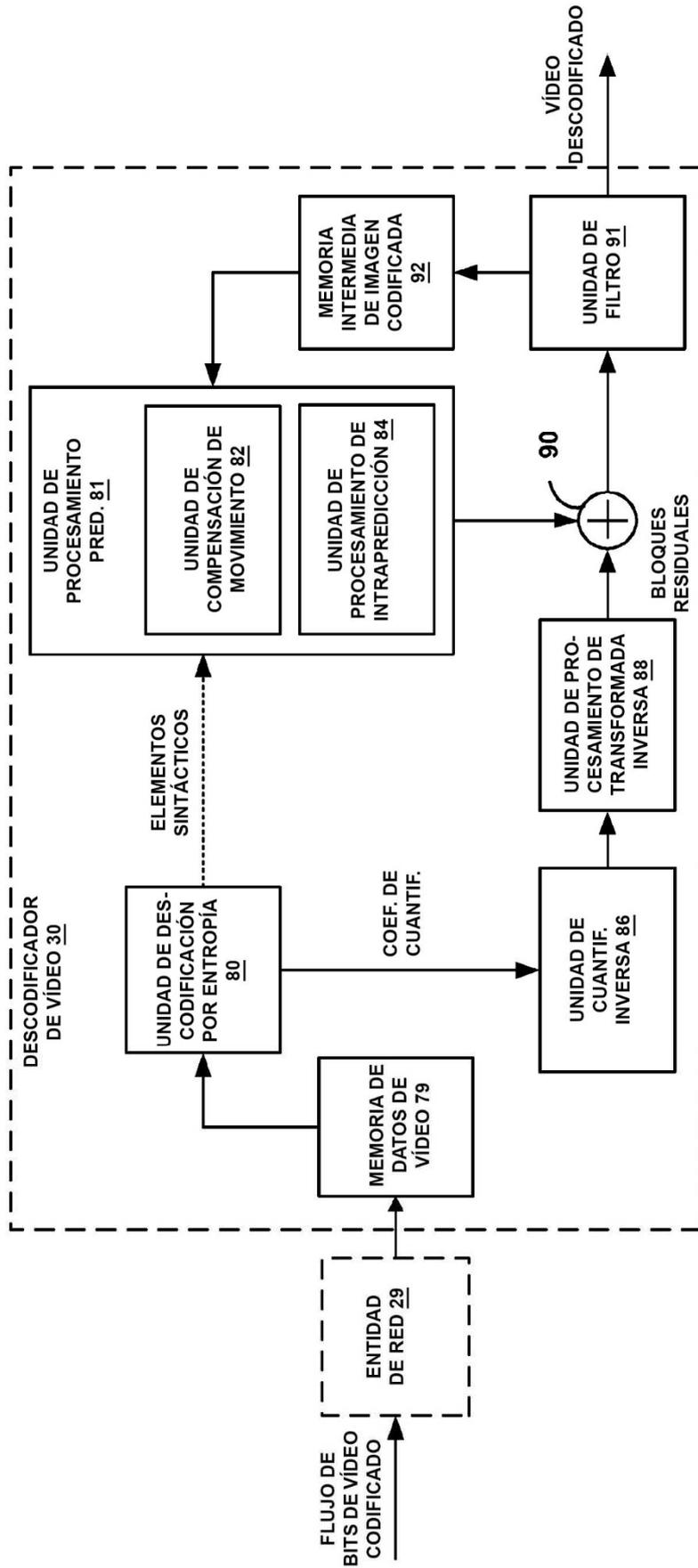


FIG. 5

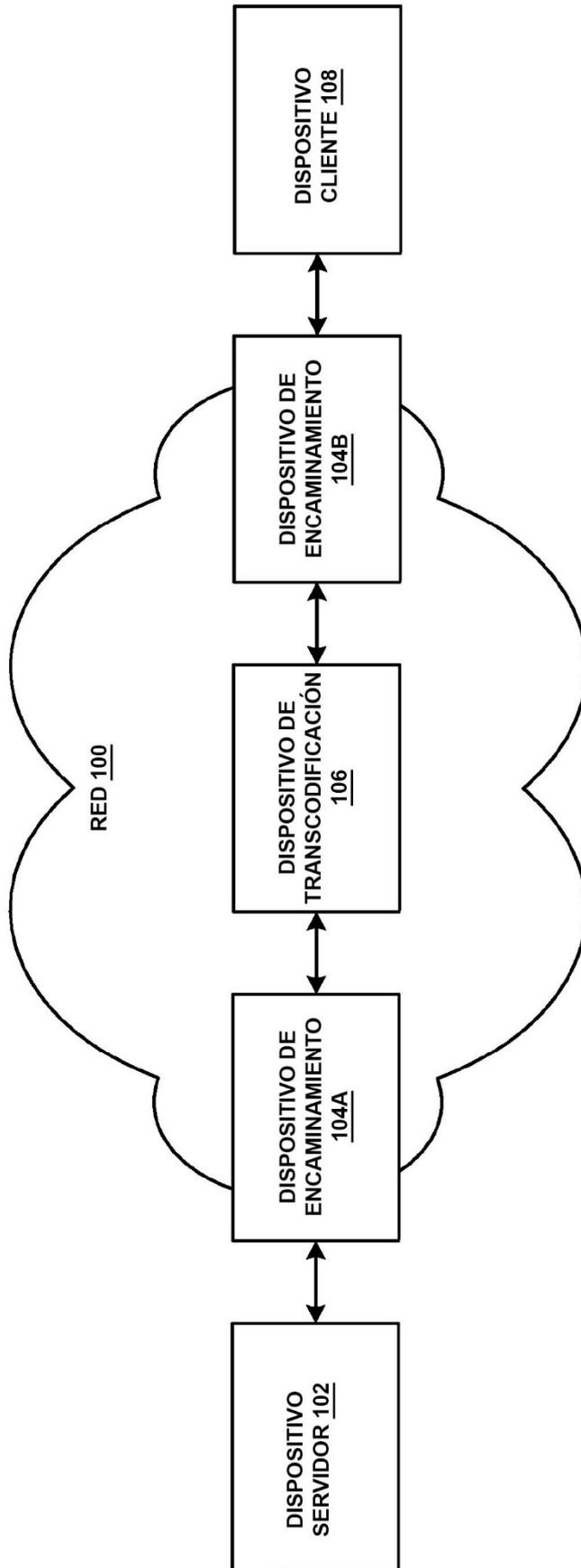


FIG. 6

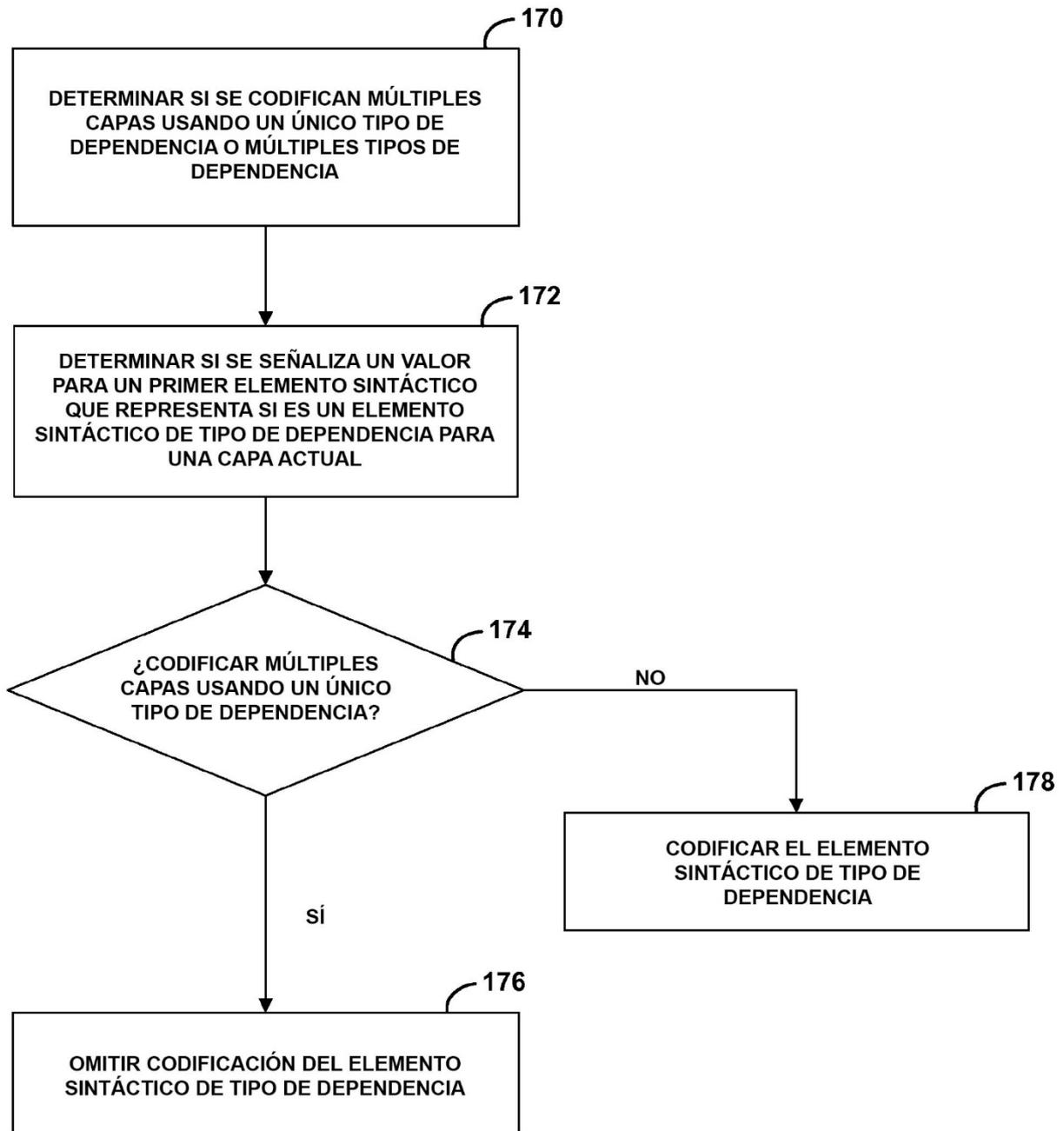


FIG. 7

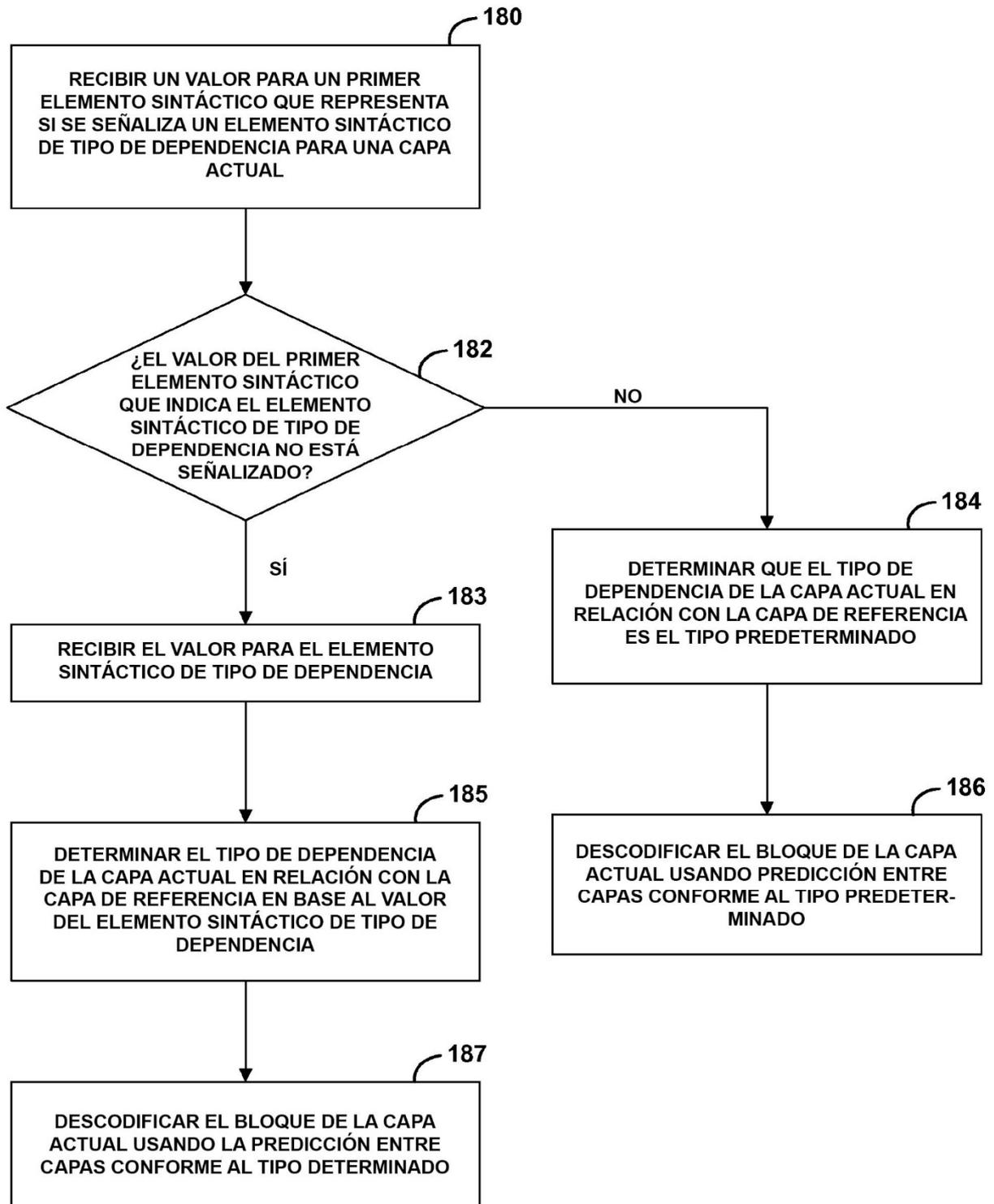


FIG. 8